이학 박사 학위논문

Variational approaches for Cauchy noise removal

(코시잡음 제거를 위한 변분법적 접근)

2020년 2월

서울대학교 대학원

수리과학부

김 건 우

Variational approaches for Cauchy noise removal

(코시잡음 제거를 위한 변분법적 접근)

지도교수 강명주

이 논문을 이학 박사 학위논문으로 제출함

2019년 10월

서울대학교 대학원

수리과학부

김 건 우

김 건 우의 이학 박사 학위논문을 인준함

2019년 12월

위 원 장 _____ (인)

부 위 원 장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

Variational approaches for Cauchy noise removal

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
to the faculty of the Graduate School of
Seoul National University

by

Geonwoo Kim

Dissertation Director : Professor Myungjoo Kang

Department of Mathematical Sciences
Seoul National University

February 2020

Abstract

In image processing, image noise removal is one of the most important problems. In this thesis, we study Cauchy noise removal by variational approaches. Cauchy noise occurs often in engineering applications. However, because of the non-convexity of the variational model of Cauchy noise, it is difficult to solve and were not studied much. To denoise Cauchy noise, we use the non-convex alternating direction method of multipliers and present two variational models. The first thing is fractional total variation(FTV) model. FTV is derived by fractional derivative which is an extended version of integer order derivative to real order derivative. The second thing is weighted nuclear norm model. Weighted nuclear norm has an excellent performance in low-level vision. We have combined our novel ideas with weighted nuclear norm minimization to achieve better results than existing models in Cauchy noise removal. Finally, we show the superiority of the proposed model from numerical experiments.

Key words: Cauchy noise, image denoising, nonconvex ADMM, fractional TV, weighted nuclear norm, variational model
Student Number: 2012-23017

Contents

Chapter 1

Introduction

Image processing is a field of computer science that refers to various forms of processing using input images. Image processing includes a variety of processes and there are no clear-cut boundaries. But image processing can be broadly classified into the following three types: low-level, min-level, and high-level processing [20]. Low-level processes include denoising, deblurring, image sharpening, contrast-enhancement etc. The characteristic of low-level vision is that both input and output data are images. Unlike low-level, mid-level process has the feature that input data are generally image but output data are attributes extracted from images. Types of output include, for example, edges, outlines, and objects inside the image. Lastly, higher-level processing are concerned with the interpretation of recognized objects or analysis of a whole image. For example, high-level vision include body pose classification, face and emotion detection, object detection and recognition.

In this thesis, we will study of image noise removal which is a part of the low-level field. Noise in image can occur during image transmission, acquisition, etc. These noise type depend on the type of disturbance in the image data. More mathematically, noise can be considered as a random variable and called as a specific noise by which random variable the noise follows. There are various noise type in image noise. For example, Gaussian noise (Amplifier noise), Salt-and-pepper noise (Impulse noise), Shot noise, Quantization noise (uniform noise), on-isotropic noise, Speckle noise (Multiplicative noise), etc [23]. Among these various noise, in this thesis, we will study

Cauchy noise which is one of the additive noise.

Cauchy noise is a type of non-Gaussian noise with additive, heavy-tailed impulsive characteristics and has appeared in atmospheric and underwater acoustic signals that are used in SAR(Synthetic Aperture Radar) and SONAR(SOund Navigation And Ranging) applications, power line communication channels, and biomedical images [11, 15, 25, 26, 27, 48, 49, 61].

Mathematically speaking, we consider following problem:

$$f = u + v$$

where u is the original image defined on a bounded domain $\Omega \subset \mathbb{R}^2$, f is degraded image, and v means Cauchy noise which follows Cauchy distribution. We want to recover u from the given noisy image f.

Recently, Cauchy noise has attracted a lot of researchers. Chang et al. [57] applied a recursive restoration algorithm based on a Markov random field to Cauchy noise removal. Loza et al. [32] used non-Gaussian statistical modelling of wavelet coefficients to noise reduction and multimodal image fusion. Afterwards, Sciacchitano et al. [45] presented the following convex TV-based variation model for Cauchy noise denoising:

$$\underset{u \in BV(\Omega)}{\arg\min} \int_\Omega |\nabla u| dx + \frac{\lambda}{2} \int_\Omega \left(\log \left(\gamma^2 + (u-f)^2 \right) dx + \mu \|u - f_0\|_F^2 \right) \tag{1.1}$$

where $\gamma > 0$ is the scale parameter that indicates the intensity of Cauchy noise and $BV(\Omega)$ is the space of the functions of bounded domain. $\lambda > 0$ is regularization parameter, which determine the proportion of the regularization term and the fidelity term. f_0 is obtained by median filtering to f. To make the convex model, they introduce the Frobenius norm term $\|u - f_0\|_F^2$. If $8\mu\gamma^2 \geq 1$, equation (1.1) is strictly convex, so, it has the unique solution. But the $\|u - f_0\|_F^2$ term pushes the solution similar to the median filter result f_0 which is far from optimal result. With the help of [52], Mei et al. [34] solved the following model without the Frobenius norm term:

$$\underset{u \in BV(\Omega)}{\arg\min} \int_\Omega |\nabla u| dx + \frac{\lambda}{2} \int_\Omega \log \left(\gamma^2 + (u-f)^2 \right) dx \tag{1.2}$$

They prove that any sequence generated by their algorithm converge to a stationary point under specific conditions.

On the other hand, Laus et al. [28] has proposed a filter based method of using myriad filters to Cauchy noise removal. They further suggested a method to measure the intensity of noise γ, which could be used to achieve better denoising results.

In this thesis, we propose two models for Cauchy noise removal. The first model uses fractional total variation. Fractional total variation derived from fractional derivative and that derivative are considered as a generalization of the ordinary derivative. We use two popular fractional derivative: the Riemann-Liouville (RL), and the Grünwald-Letnikov (GL) derivatives. RL fractional derivative derived from Cauchy formula for repeated integral and defined as follows [41]:

$$ {}_c\mathbf{D}_x^p f(x) =: \frac{1}{\Gamma(k-p)} \frac{d^k}{dx^k} \int_c^x (x-y)^{k-p-1} f(y) dy, \qquad (k-1 \leq p < k) $$

where $p > 0$ means order of derivative.

The second fractional derivative is the GL derivative denoted by

$$ {}_cD_x^p f(x) = \lim_{\substack{h \to 0 \\ n=(x-c)/h}} h^{-p} \sum_{i=0}^{n} (-1)^i \binom{p}{i} f(x-ih) $$

where $p > 0$ is order of derivative as in RL. By using these fractional derivatives, we present nonconvex fractional total variation model for Cauchy noise removal. Especially we analyze the results of each anisotropic and isotropic fractional total variation models.

The second model is a non-local method which uses weighted nuclear norm minimization [22, 21]. There are many non-local denoising methods: block matching 3D filtering (BM3D) and its improved models [13, 35, 46], non-local means (NLM) [5], non-local total variation (NLTV) [60], trilateral weighted sparse coding (TWSC) [55], and weighted nuclear norm minimization (WNNM) [22] etc. Due to the superiority and usefulness of WNNM, we applied WNNM to the Cauchy noise removal and as a result we were able to achieve better results than state-of-the-art Cauchy noise removal models.

This thesis is organized as follows. In chapter 2, we analyze Cauchy distribution and its property and Cauchy noise. Then we induce variational model of Cauchy noise by maximum a posteriori. In the last section of chapter 2, we briefly review previous work of Cauchy noise removal. In chapter 3, we define two fractional derivatives and check well-definedness of the two definitions and propose fractional total variation. We then present nonconvex ADMM algorithm with convergence analysis for our model and numerical results of the model. Finally, in chapter 4, we briefly review weighted nuclear norm and give the weighted nuclear norm model for Cauchy noise removal. Similar with chapter 3, we present nonconvex ADMM algorithm for our weighted nuclear norm model and numerical result of our model.

Chapter 2

The Cauchy distribution and the Cauchy noise

Our goal in this thesis is to denoise an additive Cauchy noise. The Cauchy noise follows the Cauchy distribution. Basically, it is necessary to understand the Cauchy noise and the Cauchy distribution in order to remove an additive Cauchy noise. In this section, we study the property of Cauchy distribution and analyze the nature and shape of Cauchy noise. Finally, we briefly review Cauchy noise removal models studied so far.

2.1 The Cauchy distribution

2.1.1 The alpha-stable distribution

The Cauchy distribution is a special case of the alpha-stable distribution. The *stable* distribution means that it is closed under addition. The exact definition is as follows [39]:

Definition 2.1.1. *A random variable X is stable if for X_1 and X_2 independent copies of X and any $a > 0$ and $b > 0$,*

$$aX_1 + bX_2 \stackrel{\circ}{=} cX + d \tag{2.1}$$

hold for some positives $c, d \in \mathbb{R}$.(The symbol $\stackrel{\circ}{=}$ means both expressions have

the same probability law.)

There is famous three cases which can be written closed form.

Example 2.1.2. *Normal distribution. For a random variable X, we write $X \sim \mathcal{N}(\mu, \sigma^2)$ if the probability density function of X is*

$$f_N(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad -\infty < x < \infty.$$

Example 2.1.3. *Cauchy distribution. We write $X \sim \mathcal{C}(x_0, \gamma)$ if the probability density function of X is*

$$f_C(x) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x-x_0)^2}, \quad -\infty < x < \infty. \tag{2.2}$$

Example 2.1.4. *Lévy distribution. We write $X \sim \text{Lévy}(\gamma, \delta)$ if the probability density function of X is*

$$f_L(x) = \sqrt{\frac{\gamma}{2\pi}} \frac{1}{(x-\delta)^{3/2}} \exp\left(-\frac{\gamma}{2(x-\delta)}\right), \quad \delta < x < \infty.$$

Normal distribution and Cauchy distribution are symmetric, bell-shaped curves but Lévy is not symmetric. Cauchy distribution has thicker tails than normal distribution. And Lévy distribution has even heavier tail than Cauchy distribution. And contrary to normal and Cauchy, Lévy distribution domain is bigger than zero. The Probability density function and Cumulative distribution function of these three densities ploted in figure 2.1.

For the two random variables X, Y, we call that X, Y have a same *type* if there exist constants $a_1 > 0$ and $a_2 \in \mathbb{R}$ with $a_1 X + a_2 \triangleq Y$. We can restate that X is stable as $aX_1 + bX_2$ has a same type with X.
There is more concrete way to describe all possible stable distribution. The probability density function of stable distribution cannot be written analytically. But, we can express the distribution by using the characteristic function or Fourier transform. (The characteristic function is defined by $\phi(u) = E[\exp(iuX)] = \int_{\infty}^{\infty} \exp(iux) dF(x)$ for a random variable X with distribution function $F(x)$.) The equivalence between 2.1.1 and 2.1.5 are given in [39].

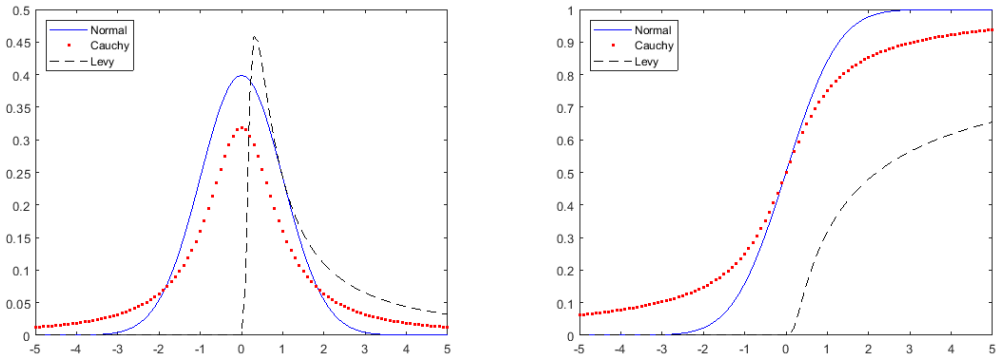


Figure 2.1: probability density function and cumulative distribution function of normal, Cauchy, Lévy distributions

Definition 2.1.5. *A random variable X is stable if $X \stackrel{\circ}{=} aZ + b$, where $0 < \alpha \leq 2, -1 \leq \beta \leq 1, a \neq 0, b \in \mathbb{R}$ and Z is a random variable with characteristic function*

$$E[\exp(iuZ)] = \begin{cases} \exp\left(-|u|^{\alpha}[1 - i\beta \tan\frac{\pi\alpha}{2}(\text{sign}(u))]\right) & \alpha \neq 1 \\ \exp\left(-|u|[1 + i\beta\frac{2}{\pi}(\text{sign}(u))\log|u|]\right) & \alpha = 1 \end{cases} \tag{2.3}$$

where sign function is

$$\text{sign}(u) = \begin{cases} -1 & u < 0 \\ 0 & u = 0 \\ 1 & u > 1. \end{cases}$$

When $\beta = 0$ and $b = 0$, the probability distribution of X is symmetric around zero.

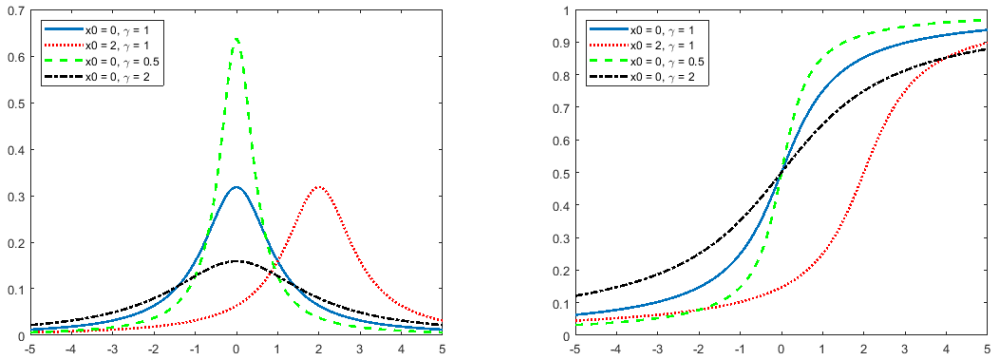
Example 2.1.6. *Let $\alpha = 2, \beta = 0, a = \sigma\sqrt{2}, b = \mu$, then the characteristic function of $X = aZ + b$ is $\exp(iu\mu - \frac{\sigma^2}{2}u^2)$. Therefore $X \sim \mathcal{N}(\mu, \sigma^2)$. And for $\alpha = 1, \beta = 0, a = \gamma, b = \delta$, the characteristic function of $X = aZ + b$ is $\exp(iu\delta - \gamma|u|)$. Therefore, X follows a Cauchy distribution $\mathcal{C}(\gamma, \delta)$.*

2.1.2 The Cauchy distribution

As seen above, Cauchy distribution is a special case of alpha-stable distribution. Cauchy distribution has two parameter x_0, γ and probability density function is given in the example 2.1.3. The cumulative distribution function is

$$F_C(x) = \frac{1}{\pi} \arctan\left(\frac{x - x_0}{\gamma}\right) + \frac{1}{2}$$

where x_0 is the location parameter which indicates the position of the peak of the distribution and γ is the scale parameter which represents the half-width at half-maximum (HWHM). Figure 2.2 shows the shape of the probability density function and cumulative distribution of Cauchy distribution according to the change of x_0 and γ.



(a) (b)

Figure 2.2: PDFs and CDFs of Cauchy distribution for some x_0, γ.

Unfortunately, mean, variance or higher moments of the Cauchy distribution has not defined. For the standard Cauchy distribution case, the mean of $\mathcal{C}(0, 1)$, if it exists, is given as follows:

$$\begin{aligned} E[X] &= \int_{-\infty}^{\infty} x f(x) dx \\ &= \int_{0}^{\infty} x f(x) dx + \int_{-\infty}^{0} x f(x) dx \end{aligned}$$

$$= \frac{1}{\pi}\int_0^\infty \frac{x}{1+x^2} + \frac{1}{\pi}\int_{-\infty}^0 \frac{x}{1+x^2}$$
$$= \frac{1}{\pi}\log(1+x^2)\Big|_0^\infty + \frac{1}{\pi}\log(1+x^2)\Big|_{-\infty}^0 \tag{2.4}$$

The two terms in (2.4) are infinity and opposite sign. Because opposite infinity sum cannot be defined, the mean of the Cauchy distribution is undefined. The mode and median are well defined. It is easy to see both values are all x_0 through the probability density function of Cauchy.

Cauchy distribution has some interesting property [17]. We present some of them in this thesis. The first proposition is stable property of the Cauchy distribution which is described in example 2.1.6. Following proposition verified by using characteristic function.

Proposition 2.1.7. *If $X \sim \mathcal{C}(a,\gamma)$ and $Y = \alpha X + \beta$ for some $\alpha, \beta \in \mathbb{R}$, then it holds $Y \sim \mathcal{C}(\alpha a+\beta, |\alpha|Y)$. Further, for independent random variables $X_1, \ldots, X_n, X_i \sim \mathcal{C}(a_i, \gamma_i)$, the relation $\sum_{i=1}^n X_i \sim \mathcal{C}(\sum_{i=1}^n a_i, \sum_{i=1}^n \gamma_i)$ is fulfilled.*

Proposition 2.1.8. *The Cauchy distribution has the inverse property. That is, if $X \sim \mathcal{C}(x_0, \gamma)$ then $Y = 1/X$ also has the Cauchy distribution.*

Proof. For the random variable $X \sim \mathcal{C}(x_0, \gamma)$, the probability density function is

$$f_X(x) = \frac{1}{\pi}\frac{\gamma}{\gamma^2 + (x-x_0)^2}.$$

With the exception of $X = 0$, the random variable Y and X have a one to one transformation $Y = g(X) = 1/X$ from $\mathcal{X} = \{x| -\infty < x < \infty\}$ to $\mathcal{Y} = \{y| -\infty < y < \infty\}$. The inverse of Jacobian is

$$\frac{dX}{dY} = -\frac{1}{Y^2}. \tag{2.5}$$

Form the (2.5), the probability density function of Y is

$$\begin{aligned} f_Y(y) &= f_X(g^{-1}(y)) \left| \frac{dx}{dy} \right| \\ &= \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (1/y - x_0)^2} \frac{1}{y^2} \\ &= \frac{1}{\pi} \frac{\gamma}{\gamma^2 y^2 + 1 - 2x_0 y + x_0^2 y^2} \\ &= \frac{1}{\pi} \frac{\gamma}{(\gamma^2 + x_0^2) y^2 - 2x_0 y + \frac{x_0^2}{\gamma^2 + x_0^2} + \frac{\gamma^2}{\gamma^2 + x_0^2}} \\ &= \frac{1}{\pi} \frac{\frac{\gamma}{\gamma^2 + x_0^2}}{\frac{\gamma^2}{(\gamma^2 + x_0^2)^2} + \left(y - \frac{x_0}{\gamma^2 + x_0^2} \right)^2} \end{aligned}$$

which is the probability density fuction of the Cauchy distribution. Therefore,

$$Y \sim \mathcal{C}\left(\frac{x_0}{\gamma^2 + x_0^2}, \frac{\gamma}{\gamma^2 + x_0^2} \right).$$

□

Following proposition said the relation between normal distribution and Cauchy distribution. This proposition can be used for generating Cauchy distribution.

Proposition 2.1.9. *Let X and Y be two independent random variables such that $X \sim \mathcal{N}(0, \sigma_x^2)$ and $Y \sim \mathcal{N}(0, \sigma_y^2)$. Then for the random variable $Z = X/Y$, $Z \sim \mathcal{C}(0, \frac{\sigma_x^2}{\sigma_y^2})$.*

Proof. The joint distribution of X and Y has the following probability density function:

$$f(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}}.$$

From [12], the probability density function of the distribution of $Z = X/Y$

is following:

$$\begin{aligned} p(z) &= \int_{-\infty}^{\infty} |y| f(zy, y) dy = \frac{1}{\pi \sigma_x \sigma_y} \int_0^{\infty} y e^{-y^2 \left(\frac{z^2}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \right)} dy \\ &= -\frac{1}{2\left(\frac{z^2}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \right)} \exp\left(-y^2 \left(\frac{z^2}{2\sigma_x^2} + \frac{1}{2\sigma_y^2} \right) \right) \Bigg|_0^{-\infty} \\ &= \frac{1}{\pi \sigma_x \sigma_y} \frac{1}{\frac{z^2}{\sigma_x^2} + \frac{1}{\sigma_y^2}} = \frac{1}{\pi} \frac{\sigma_x / \sigma_y}{z^2 + (\sigma_x / \sigma_y)^2} \end{aligned}$$

Therefore, according to the (2.2), $p(z)$ is the probability density function of Cauchy distribution with location parameter 0 and scale parameter σ_x/σ_y. □

A distribution is said to be right *heavy-tailed*, if

$$\lim_{x \to \infty} e^{\lambda x} \mathbb{P}(X > x) = \infty \quad \text{for} \quad \forall \lambda > 0 \tag{2.6}$$

and left heavy-tailed also can be defined similarly.
According to (2.6), Cauchy distribution is heavy-tailed. Laplace distribution is another famous heavy-tailed distribution. Laplace distribution is defined as follows

Definition 2.1.10. *A random variable X has a Laplace distribution if its probability density function is*

$$f_{Lap}(x) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b} \right) \tag{2.7}$$

and we write $X \sim \text{Lap}(\mu, b)$.

The cumulative distribution function of Laplace distribution is as follows:

$$F_{Lap}(x) = \frac{1}{2} + \frac{1}{2} \text{sign}(x - \mu) \left(1 - \exp\left(-\frac{|x - \mu|}{b} \right) \right).$$

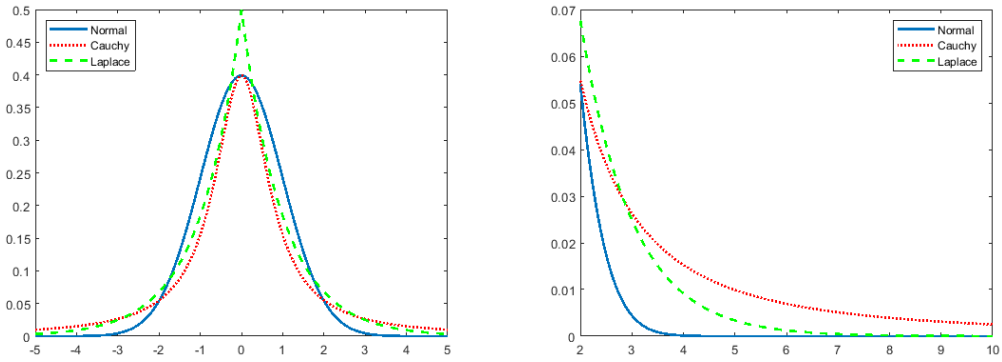
Both the Laplacian and Cauchy distribution are heavy-tailed distribution. But the actual thickness is different. We can compare thickness of three

distributions.[34]

Proposition 2.1.11. *Let f_N, f_C and f_{Lap} denote the probability density functions for $\mathcal{N}(0,\sigma^2), \mathcal{C}(0,\sqrt{\frac{2}{\pi}}\sigma)$ and $\mathrm{Lap}(0,\sigma)$ respectively. Then, the followings hold:*

1. *At $x = 0$, $f_N = f_C$.*
2. *At $x = \sigma$, the ratio of f_N, f_{Lap} and f_C is $1 : \sqrt{\frac{\pi}{2e}} : \sqrt{\frac{e}{2\pi}}$;*
3. *At $x = 3\sigma$, the ratio of f_N, f_{Lap} and f_C is $1 : \sqrt{\frac{\pi}{2}}e^{3/2} : \sqrt{\frac{1}{50\pi}}e^{9/2}$.*

Based on Proposition 2.1.11, we can see that the probability density value of the normal distribution at a rather small x, saying $x = \sigma$, is the largest. However, at rather large point x, saying $x = 3\sigma$, the density of the Laplace distribution is more than 5 times that of the normal distribution, and the density of the Cauchy distribution is even more than 7 times.



(a)The plots of the three distributions (b) the zoomed-in $[2, 10]$.

Figure 2.3: PDFs of normal($\mathcal{N}(0,1)$), Cauchy($\mathcal{C}(0,\sqrt{\frac{2}{\pi}})$ and Laplace ($\mathrm{Lap}(0,1)$) distributions.

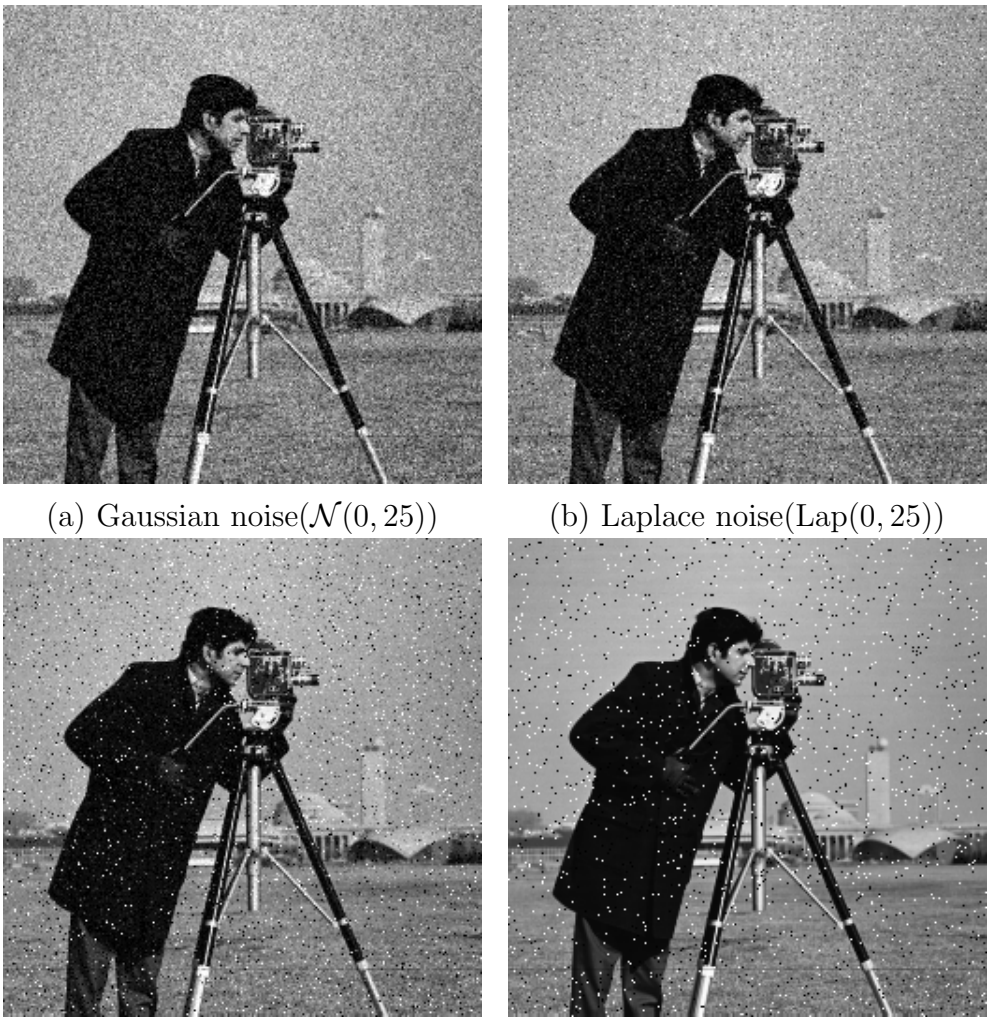
In figure 2.3, we present the probability density functions of normal, Laplace and Cauchy distributions. From figure 2.3 (a), the normal distribution has the same peak value with the Cauchy distribution and the normal distribution is slightly higher on near the peak. At the tail (figure 2.3

(b)), however, the Cauchy distribution is highest, followed by the Laplace distribution and the normal distribution.

2.2 The Cauchy noise

2.2.1 Analysis of the Cauchy noise

For image deblurring and denoising, most of the research consider the restoration problem which is corrupted by additive Gaussian noise [8, 19, 60, 54, 44]. But in many image processing application, there is very impulsive noise problem exist. One such kind of noise is Cauchy noise, which has been relatively less studied. Cauchy noise appears in atmospheric and under water acoustic noise, for example, images obtained by sound navigation and ranging (SONAR) or synthetic aperture radar (SAR) images, wireless communication systems, biomedical images [34, 45, 25, 43, 40]. As shown in figure 2.3, the Cauchy distribution tends to have a thicker tail than the normal or Laplace distribution. We compare this thickness phenomenon by noisy image. From the figure 2.4, we can see Gaussian noisy image and Laplace noisy image are not much different. Actually, in the Laplace noisy image some pixels are more heavily contaminated than the Gaussian noisy image, but the Cauchy noise has much more impulsive noisy pixel than the Gaussian and Laplace noisy image. While in some way, Cauchy noisy image is quite close to salt and pepper noisy image. For example, Cauchy noise and salt and pepper noise have some pixels contaminated with white or black. This pattern is not seen in Gaussian and Laplace noise. On the other hand, there are some difference between Cauchy and salt and pepper noise. For instance, in the salt and pepper noisy image has some noise-free pixels, while all the pixels are corrupted by noise in the Cauchy noisy image. In summary, the Cauchy noise is similar to a mixture of Gaussian noise and salt and pepper noise.



(a) Gaussian noise($\mathcal{N}(0, 25)$) (b) Laplace noise(Lap$(0, 25)$)

(c) Cauchy noise($\mathcal{C}(0, 3)$) (d) salt and pepper noise(3%)

Figure 2.4: Noisy images comparision.

2.2.2 Variational model of Cauchy noise

In this subsection, we find variational model of Cauchy noise denoising. For the Cauchy noisy image f defined on $\Omega \subset \mathbb{R}^2$, the clean image u follows the equation:

$$f = u + v$$

where v is the Cauchy noise. Similar with [34, 45], we get variational model using the Bayes theorem and the maximum a posteriori(MAP) estimation. In the following, we write random variables as the capital letter F, U, V and the respective probability density functions as $P(F), P(U), P(V)$.
As in most denoising research, we naturally assume Cauchy noise random variable V follows zero-contered distribution $V \sim \mathcal{C}(0, \gamma)$, i.e., $f_C(v) = \frac{\gamma}{\pi(\gamma^2+v^2)}$.

Using MAP, we want to find clean image U by maximizing the conditional probability $P(U|F)$ for given noisy image F. From Bayes' theorem, we can get the follows:

$$\arg\max_U P(U|F) = \arg\max_U \frac{P(F|U)P(U)}{P(F)}. \tag{2.8}$$

We take logarithm to remove fraction and take minus to change maximum to minimum then (2.8) is changed to

$$\begin{aligned}\arg\min_U -\log(P(U|F)) &= \arg\min_U -(\log(P(F|U)) + \log P(U) - \log P(F)) \\ &= \arg\min_U -(\log(P(F|U)) + \log P(U)). \end{aligned} \tag{2.9}$$

In the above, because the $\log P(F)$ term is independent of random variable U, we drop the term. For an pixel $x \in \Omega$, with Ω the domain of the image, we have

$$P(f(x)|u(x)) = \frac{\gamma}{\pi(\gamma^2 + (u(x) - f(x))^2)}. \tag{2.10}$$

Using the property that the pixels $x \in \Omega$ are mutually independent and identically distributed, we can get $P(U) = \prod_{x\in\Omega} P(U(x))$. Therefore, (2.9) is equivalent to the following:

$$\begin{aligned}&-\arg\min_U \log(P(U|F)) \\ &= -\arg\min_U \int_\Omega (\log(P(F(x)|U(x))) + \log(P(U(x)))) \, dx\end{aligned}$$

$$= \arg\min_u \int_\Omega \left(\log(\gamma^2 + (u(x) - f(x))^2) + \log(\pi) - \log(\gamma) - \log(P(u(x)))\right)$$
$$= \arg\min_u \int_\Omega \left(\log(\gamma^2 + (u(x) - f(x))^2) - \log(P(u(x)))\right). \quad (2.11)$$

In the second equation, we use (2.10) and in the third equation, we remove constant term.
To compute $\log(P(u(x)))$ term, we need a priori. In [45, 34], they assume that U follows a Gibbs prior:

$$P(u) = \frac{\exp(-\beta J(u))}{Z} \quad (2.12)$$

where β is positive number and Z is normalizing factor and J is non-negative function given by an special feature of natural images. In this thesis, we assume U follows a Gibbs prior and apply some functions to J that are effectively used to remove Gaussian noise denoising.

2.3 Previous work

In recent years, many researchers have been interested in Cauchy noise denoising and proposed several novel model. According to [1], Achim et. al. designed a bivariate maximum a posteriori estimator to capture the heavy-tailed densities. They modeled the interscale dependency of wavelet coefficient which is existed in noisy image and estimate the parameters of the aplha-stable distribution based on the noisy observation. In [45], Sciacchitano et. al. propose the following convex Cauchy noise denoising model:

$$\arg\min_{u \in BV(\Omega)} \int_\Omega |Du| + \frac{\lambda}{2}\left(\int_\Omega \log\left(\gamma^2 + (u - f)^2\right) dx + \mu\|u - u_0\|_F^2\right), \quad (2.13)$$

where u_0 is the image obtained by applying the median filter to the noisy image f and $\lambda > 0$ and $\mu > 0$ are the regularization parameters. Due to the presence of the log, the (2.13) is generally not a convex model. Because of the $\|u - u_0\|_F^2$ term, however, they proved that under some conditions, the (2.13) becomes a convex model. The theorem is as follows:

Theorem 2.3.1. *If* $8\mu\gamma^2 \geq 1$*, the model defined in* (2.13) *is strictly convex. Therefore there exists a unique solution for the minimization problem.*

Because the convergence of Chambolle -Poce algorithm is guaranteed by [9], they used Chambolle-Pock algorithm for numerical method. This model was novel at solving the Cauchy noise model by introducing the median filter term. But median filter term pushes the solution similar to the median filter result.

After the paper published, some researchers have proved some precious convergence porperties for solving nonconvex minimization problems [50, 52, 56]. By making use of the results, Mei et. al. proposed a nonconvex TV-based variational model for denoising Cauchy noise.

$$\arg\min_{u\in BV(\Omega)} \int_\Omega |Du| + \frac{\lambda}{2}\int_\Omega \log(\gamma^2 + (u-f)^2)dx \tag{2.14}$$

And they develop the nonconvex alternating direction method of multipliers(ADMM) algorithm to solve the (2.14). They prove that the sequence generated by ADMM algorithm of (2.14) converges globally to a stationary point of augmented Lagrangian of (2.14).

In 2018, Laus et. al. [28] proposed new method for Cauchy noise removal which is based on the myriad filter. They collect similar pixels through local or nonlocal matching. The local matching based on the assumption that the small neighborhood pixels have the almost same values. The non-local approach use the self-similarity assumption of the image. For each pixel index $x \in \Omega$, they collect similar pixels $\mathcal{A}(x)$ and estimate clean image pixel by following maximum likely-hood estimation:

$$U^* \in \arg\min_{U,\gamma} \sum_{x\in\Omega} \sum_{y\in\mathcal{A}(x)} \log\left((F(y)-U(x))^2 + \gamma^2\right) - \log(\gamma) \tag{2.15}$$

Also they propose noise intensity γ estimation. By solving the (2.15), they shown better performance than the total variation based models.

Chapter 3

Fractional order derivatives and total fractional order variational model

In this chapter we first look at two famous fractional order derivatives (Grünwald-Letnikov and Riemann-Liouville) and their properties. Next, we will apply the fractional order derivatives to the Cauchy noise denoising model and introduce the nonconvex alternating direction method of multipliers which is used to solve our model. Finally we examine the numerical results of our fractional total variation model.

3.1 Some fractional derivatives and integrals

Fractional order derivatives can be considered as a generalization of the integer order derivatives to the real order or complex order. Because the generalization is not unique, there are several fractional derivatives exist. Among them, we examine two popularly used fractional derivatives: the Grünwald-Letnikov and the Riemann-Liouville.

3.1.1 Grünwald-Letnikov Fractional Derivatives

Grünwald-Letnikov derivative basically extended from the natural derivative. Assume $y = f(x)$ is a continuous function. By the ordinary definition, first order derivative of $f(x)$ is defined as follows:

$$f'(x) = \frac{df}{dx} = \lim_{h\to 0} \frac{f(x) - f(x-h)}{h}$$

by applying derivative operator again, we get the second-order derivative:

$$\begin{aligned} f''(x) = \frac{d^2 f}{dx^2} &= \lim_{h\to 0} \frac{f'(x) - f'(x-h)}{h} \\ &= \lim_{h\to 0} \frac{f(x) - 2f(x-h) + f(x-2h)}{h^2}. \end{aligned}$$

and apply derivative operator again, we obtain:

$$f'''(x) = \frac{d^3 f}{dx^3} = \lim_{h\to 0} \frac{f(x) - 3f(x-h) + 3f(x-2h) - f(x-3h)}{h^3}$$

by the help of the induction,

$$f^{(n)} = \frac{d^n f}{dx^n} = \lim_{h\to 0} \frac{1}{h^n} \sum_{r=0}^{n} (-1)^r \binom{n}{r} f(x - rh), \tag{3.1}$$

where

$$\binom{n}{r} = \frac{n(n-1)(n-2)\dots(n-r+1)}{r!}.$$

is the binomial coefficient. Now we generalize the (3.1) to integer order:

$$f_h^{(p)}(x) = \frac{1}{h^p} \sum_{i=0}^{n} (-1)^k \binom{p}{i} f(x - ih) \tag{3.2}$$

where p is an arbitrary integer number. For $0 \le p \le n$, we have

$$\lim_{h\to 0} f_h^{(p)}(x) = f^{(p)}(x) = \frac{d^p f}{dx^p},$$

because $\binom{p}{i}$ for $i > p$ are equal to 0.
Next, we consider $p < 0$. Let us denote

$$\begin{bmatrix} p \\ r \end{bmatrix} = \frac{p(p+1)\cdots(p+r-1)}{r!}.$$

Then we can easily check

$$\binom{-p}{r} = (-1)^r \begin{bmatrix} p \\ r \end{bmatrix}.$$

For convenience, we replace p in (3.2) with $-p$ then (3.2)

$$f_h^{(-p)}(x) = \frac{1}{h^p} \sum_{i=0}^{n} \begin{bmatrix} p \\ i \end{bmatrix} f(x - ih) \tag{3.3}$$

where p is a positive integer.
If n is fixed, then $\lim_{h\to 0} f_h^{(-p)}(x) = 0$. For a real number c, let us consider $n = \frac{x-c}{h}$. Then define

$$\lim_{\substack{h\to 0 \\ n=(x-c)/h}} f_h^{(-p)}(x) := \ {}_cD_x^{-p} f(x)$$

If $p = 1$,

$$f_h^{(-1)}(x) = h \sum_{i=0}^{n} f(x - ih).$$

Since f is continuous function, we get

$$\lim_{\substack{h\to 0 \\ n=(x-c)/h}} f_h^{(-1)}(x) = \int_0^{x-c} f(x-y)dy = \int_c^x f(y)dy.$$

According to [41], we can get the general formula for a positive integer p.

Proposition 3.1.1. *Let us assume that f is a continuous function. For an positive integer p, the following equation is satisfied:*

$$ {}_cD_x^{-p}(x) = \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^p \sum_{i=0}^{n} \begin{bmatrix} p \\ i \end{bmatrix} f(x-ih) = \frac{1}{(p-1)!}\int_c^x (x-y)^{p-1} f(y)dy \tag{3.4} $$

Proof. Prove by induction. Let us assume that (3.4) holds for some p. We already shown (3.4) satisfied for $p = 1$, so, define

$$ f_1(x) := \lim_{\substack{h\to 0 \\ n=(x-c)/h}} f_h^{(-1)}(x) = \int_c^x f(y)dy. \tag{3.5} $$

For the case $p + 1$,

$$ \begin{aligned} {}_cD_x^{-p-1} f(x) &= \lim_{\substack{h\to 0 \\ n=(x-c)/h}} \sum_{i=0}^{n} \begin{bmatrix} p+1 \\ i \end{bmatrix} f(x-ih) \\ &= \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^p \sum_{i=0}^{n} \begin{bmatrix} p+1 \\ i \end{bmatrix} f_1(x-ih) \\ &\quad - \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^p \sum_{i=0}^{n} \begin{bmatrix} p+1 \\ i \end{bmatrix} f_1(x-(i+1)h) \end{aligned} \tag{3.6} $$

Then by using the relation

$$ \begin{bmatrix} p+1 \\ i \end{bmatrix} = \begin{bmatrix} p \\ i \end{bmatrix} + \begin{bmatrix} p+1 \\ i-1 \end{bmatrix} $$

to the first sum and substituting i for $i + 1$ to the second sum in (3.6).

$$ \begin{aligned} {}_cD_x^{p-1} f(x) &= \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^p \sum_{i=0}^{n} \begin{bmatrix} p \\ i \end{bmatrix} f_1(x-ih) + \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^p \sum_{i=0}^{n} \begin{bmatrix} p+1 \\ i-1 \end{bmatrix} f_1(x-ih) \\ &\quad - \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^p \sum_{i=0}^{n+1} \begin{bmatrix} p+1 \\ i-1 \end{bmatrix} f_1(x-ih) \end{aligned} \tag{3.7} $$

$$
\begin{aligned}
&= {}_cD_x^{-p} f_1(x) - \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^p \begin{bmatrix} p+1 \\ n \end{bmatrix} f_1(x-(n+1)h) \\
&= {}_cD_x^{-p} f_1(x) - (x-c)^p \lim_{n\to\infty} \begin{bmatrix} p+1 \\ n \end{bmatrix} \frac{1}{n^p} f_1(c - \frac{x-c}{n}). \qquad (3.8)
\end{aligned}
$$

Using the limit definition of the gamma function [41], we derive

$$
\frac{1}{\Gamma(p+1)} = \lim_{n\to\infty} \frac{(p+1)(p+2)\cdots(p+n)}{n^p n!} = \lim_{n\to\infty} \begin{bmatrix} p+1 \\ n \end{bmatrix} \frac{1}{n^p}. \qquad (3.9)
$$

Apply (3.9) and property $f_1(c) = \lim_{n\to\infty} f_1(c-(x-c)/n) = 0$ to (3.8) then,

$$
\begin{aligned}
{}_cD_x^{-p-1} f(x) &= {}_cD_x^{-p} f_1(x) - \frac{1}{(p-1)!} \int_c^x (x-y)^{p-1} f_1(y) dy \\
&= \frac{1}{p!} \int_c^x (x-y)^p f(y) dy
\end{aligned}
$$

which means that the (3.4) is satisfied for $p+1$. □

From (3.4),

$$
\begin{aligned}
\frac{d}{dx} \left({}_cD_x^{-p} f(x) \right) = \frac{1}{(p-2)!} \int_c^x (x-y)^{p-2} f(y) dy &= {}_cD_x^{-p+1} f(x) \\
{}_cD_x^{-p} f(x) &= \int_c^x \left({}_cD_x^{-p+1} f(y) \right) dy
\end{aligned}
$$

so,

$$
{}_cD_x^{-p} f(x) = \underbrace{\int_c^x \cdots \int_c^x}_{p \text{ times}} f(x) \underbrace{dx \cdots dx}_{p \text{ times}}.
$$

Therefore, the (3.4) is just a p-fold integral.
In summary, the following expression

$$
{}_cD_x^p f(x) = \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^{-p} \sum_{i=0}^{n} (-1)^i \binom{p}{i} f(x-ih) \qquad (3.10)
$$

is the p-th derivative for positive integer p or the $-p$-fold integral for negative

integer p.

Until now, we extend ordinary derivative to integer order. Now, let us extend p to the real number. First, we consider of $p < 0$. We have to check the existence of the (3.14). To do this, we need following theorem in [41].

Theorem 3.1.2. *For given sequences $\alpha_{n,k}, \beta_k$, $(k = 1, 2, \dots)$ and suppose that*

$$\lim_{k\to\infty} \beta_k = 1,$$
$$\lim_{n\to\infty} \alpha_{n,k} = 0 \quad \text{for all } k,$$
$$\lim_{n\to\infty} \sum_{k=1}^{n} \alpha_{n,k} = A$$
$$\sum_{k=1}^{n} |\alpha_{n,k}| < K \quad \text{for all } n,$$

Then,

$$\lim_{n\to\infty} \sum_{k=1}^{n} \alpha_{n,k}\beta_k = A.$$

To apply Theorem 3.1.2 to (3.14), we substitute $p < 0$ as $-p(p > 0)$ and rewritten,

$$\begin{aligned}
{}_aD_x^{-p} f(x) &= \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^p \sum_{i=0}^{n} \begin{bmatrix} p \\ i \end{bmatrix} f(x - ih) \\
&= \lim_{\substack{h\to 0 \\ n=(x-c)/h}} \sum_{i=0}^{n} \frac{1}{i^{p-1}} \begin{bmatrix} p \\ i \end{bmatrix} h(ih)^{p-1} f(x - ih) \\
&= \frac{1}{\Gamma(p)} \lim_{n\to\infty} \sum_{i=0}^{n} \frac{\Gamma(p)}{i^{p-1}} \begin{bmatrix} p \\ i \end{bmatrix} \frac{x-c}{n} \left(i\frac{x-c}{n} \right)^{p-1} f(x - i\frac{x-c}{n}).
\end{aligned}$$

Then set

$$\alpha_{n,i} = \frac{x-c}{n}\left(i\frac{x-c}{n}\right)^{p-1} f(x - i\frac{x-c}{n})$$
$$\beta_i = \frac{\Gamma(p)}{i^{p-1}}\begin{bmatrix} p \\ i \end{bmatrix}.$$

From the limit definition of the Gamma function (3.9),

$$\lim_{i\to\infty} \beta = 1$$

And

$$\begin{aligned}\lim_{n\to\infty}\sum_{i=0}^{n}\alpha_{n,i} &= \lim_{h\to 0}\sum_{i=0}^{n} h(ih)^{p-1} f(t-ih) \\ &= \int_c^x (x-y)^{p-1} f(y)dy.\end{aligned}$$

Therefore, using theorem 3.1.2, we can get the limit value of the (3.14):

$${}_cD_x^{-p} f(x) = \frac{1}{\Gamma(p)}\int_c^x (x-y)^{p-1} f(y)dy. \tag{3.11}$$

If $f(x)$ has $k+1$-th continuous derivative (i.e. f is $\mathcal{C}^{k+1}$ function), then

$${}_cD_x^{-p} f(x) = \sum_{i=0}^{k}\frac{f^{(i)}(c)(x-c)^{p+i}}{\Gamma(p+i+1)} + \frac{1}{\Gamma(p+i+1)}\int_c^x (x-y)^{p+k} f^{(k+1)}(y)dy$$

from integration by parts.

Next, we check the existence of the (3.14) for the case $p > 0$. For convenience, let us define

$$f_h^{(p)}(x) = h^{-p}\sum_{i=0}^{n}(-1)^i\binom{p}{i} f(x-ih).$$

By the recursive relation of the binomial coefficients, we can write

$$\begin{aligned} f_h^p(x) &= h^{-p}\sum_{i=0}^{n}(-1)^i\binom{p-1}{i}f(x-ih) + h^{-p}\sum_{i=1}^{n}(-1)^r\binom{p-1}{r-1}f(x-ih) \\ &= (-1)^n\binom{p-1}{n}h^{-p}f(c) + h^{-p}\sum_{i=0}^{n-1}(-1)^i\binom{p-1}{i}\Delta f(x-ih). \end{aligned}$$

where $\Delta f(x-ih) = f(x-ih) - f(x-(i+1)h)$. Again, we use the recursive relation of the binomial coefficients m times,

$$\begin{aligned} f_h^{(p)}(x) &= (-1)^n\binom{p-1}{n}h^{-p}f(c) + (-1)^{n-1}\binom{p-2}{n-1}h^{-p}\Delta f(c+h) \\ &\quad + h^{-p}\sum_{i=0}^{n-2}(-1)^i\binom{p-2}{i}\Delta^2 f(x-ih) \\ &= \sum_{k=0}^{m}(-1)^{n-k}\binom{p-k-1}{n-k}h^{-p}\Delta^k f(c+kh) \\ &\quad + h^{-p}\sum_{i=0}^{n-m-1}(-1)^i\binom{p-m-1}{i}\Delta^{m+1}f(x-ih). \end{aligned} \tag{3.12}$$

By using the relation $x-c = nh$ and (3.9), the limit of the k-th term in the frist sum in (3.12),

$$\begin{aligned} &\lim_{\substack{h\to 0 \\ nh=x-c}} (-1)^{n-k}\binom{p-k-1}{n-k}h^{-p}\Delta^k f(c+kh) \\ &\qquad = (x-c)^{-p+k}\lim_{n\to\infty}(-1)^{n-k}\binom{p-k-1}{n-k}(n-k)^{p-k} \\ &\qquad \times \lim_{n\to\infty}\left(\frac{n}{n-k}\right)^{p-k} \times \lim_{h\to 0}\frac{\Delta^k f(c+kh)}{h^k} \\ &\qquad = \frac{(x-c)^{-p+k}f^{(k)}(c)}{\Gamma(-p+k+1)}. \end{aligned}$$

We use theorem 3.1.2 to find limit value of the second sum in (3.12). Let us

define

$$\alpha_{n,i} = h(ih)^{m-p}\frac{\Delta^{m+1}f(x-ih)}{h^{m+1}}, \quad h = \frac{x-c}{n}$$
$$\beta_i = (-1)^i\Gamma(-p+m+1)\binom{p-m-1}{i}r^{-m+p}$$

Then we can easily check $\lim_{i\to\infty}\beta_i = 1$. And for $m-p+1>0$,

$$\lim_{\substack{n\to\infty \\ nh=x-c}}\sum_{i=0}^{n-m-1}\alpha_{n,i} = \int_c^x (x-y)^{m-p}f^{(m+1)}(y)dy.$$

The limit value of the second sum in (3.12) is

$$\begin{aligned}
&\lim_{\substack{n\to\infty \\ nh=x-c}} h^{-p}\sum_{i=0}^{n-m-1}(-1)^i\binom{p-m-1}{i}\Delta^{m+1}f(x-ih) \\
&= \lim_{\substack{n\to\infty \\ nh=x-c}}\frac{1}{\Gamma(-p+m+1)}\sum_{i=0}^{n-m-1}(-1)^i\Gamma(-p+m+1)\binom{p-m-1}{i} \\
&\quad i^{-m+p}h(ih)^{m-p}\frac{\Delta^{m+1}f(x-ih)}{h^{m+1}} \\
&= \frac{1}{\Gamma(-p+m+1)}\int_c^x (x-y)^{m-p}f^{(m+1)}(y)dy
\end{aligned}$$

Therefore, we get the following conclusion:

$$\begin{aligned}
{}_cD_x^p f(x) = \lim_{\substack{n\to\infty \\ nh=x-c}} f_h^{(p)}(x) &= \sum_{i=0}^{m}\frac{f^{(k)}(c)(x-c)^{-p+k}}{\Gamma(-p+k+1)} \\
&+ \frac{1}{\Gamma(-p+m+1)}\int_c^x (x-y)^{m-p}f^{(m+1)}(y)dy. \qquad (3.13)
\end{aligned}$$

Note that the (3.13) holds for $f\in\mathcal{C}^{m+1}([c,x])$ and $m+1>p$.
Finally, we checked the well-definedness of the (3.14). Now we give formal definition of the Grünwald-Letnikov fractional derivatives.

Definition 3.1.3. *For an real number p, Grünwald-Letnikov fractional derivative is defined as follows:*

$$ {}_cD_x^p f(x) = \lim_{\substack{h\to 0 \\ n=(x-c)/h}} h^{-p} \sum_{i=0}^{n} (-1)^i \binom{p}{i} f(x-ih) \tag{3.14} $$

Next section, we will study the Riemann-Liouville fractional derivative.

3.1.2 Riemann-Liouville Fractional Derivatives

Riemann-Liouville(RL) fractional derivative starts with Cauchy formula for repeated integration [16].

Theorem 3.1.4. *Let us f is continuous function on $\mathbb{R}$. The Cauchy formula says that n-th repeated integral of f based on c is rewritten as follows:*

$$ f^{(-n)}(x) = \frac{1}{\Gamma(n)} \int_c^x (x-y)^{n-1} f(y) dy. \tag{3.15} $$

Proof. We will prove by induction. For the case $n = 1$, the equation (3.15) trivially holds. Suppose (3.15) is true for n, then using the Leibniz integral rule,

$$ \begin{aligned} f^{-n-1}(x) &= \int_c^x \frac{1}{\Gamma(n)} \left(\int_c^z (z-y)^{n-1} f(y) dy \right) dz \\ &= \int_c^x \frac{1}{\Gamma(n+1)} \left(\frac{d}{dz} \int_c^z (z-y)^n f(y) dy \right) dz \\ &= \frac{1}{\Gamma(n+1)} \int_c^x \frac{d}{dz} \left(\int_c^z (z-y)^n dz \right) f(y) dy \\ &= \frac{1}{\Gamma(n+1)} \int_c^x (x-y)^n f(y) dy \end{aligned} $$

Therefore the (3.15) holds for $n + 1$. □

Now extend the integer integrals to integrals of arbitrary order. We re-

place the integer n in (3.15) to a real $p > 0$:

$$ {}_c\mathbf{D}_x^{-p} f(x) := \frac{1}{\Gamma(p)} \int_c^x (x-y)^{p-1} f(y) dy. \tag{3.16} $$

Naturally, we expect the following equation holds:

$$ \lim_{p \to 0} {}_c\mathbf{D}_x^{-p} f(x) = f(x) \tag{3.17} $$

We can easily show that the relationship (3.17) is satisfied for $f \in \mathcal{C}^1(\mathbb{R})$. In that case,

$$ {}_c\mathbf{D}_x^{-p} f(x) = \frac{(x-c)^p f(c)}{\Gamma(p+1)} + \frac{1}{\Gamma(p+1)} \int_c^x (x-y)^p f'(y) dy $$

where we used integration by parts and we take limit

$$ \lim_{p \to 0} {}_c\mathbf{D}_x^{-p} f(x) = f(c) + \int_c^x f'(y) dy = f(x). $$

Even if $f(x)$ is only continuous for $x \geq c$, equation (3.17) holds. See [41] for the proof of continuous for $x \geq c$ case.

Therefore, we can put

$$ {}_c\mathbf{D}_x^0 f(x) = f(x). $$

Next, we derive following property of composition of operator $\mathbf{D}$.

Proposition 3.1.5. *If $f(x)$ is continuous for $x \geq c$, $p > 0$ and $q > 0$, then the integration of real order ${}_c\mathbf{D}_x$ has the following property:*

$$ {}_c\mathbf{D}_x^{-p} \left({}_c\mathbf{D}_x^{-q} f(x) \right) = {}_c\mathbf{D}_x^{-p-q} f(x). \tag{3.18} $$

Proof.

$$ {}_c\mathbf{D}_x^{-p} \left({}_c\mathbf{D}_x^{-q} f(x) \right) = \frac{1}{\Gamma(p)} \int_c^x (x-y)^{p-1} \, {}_c\mathbf{D}_y^{-q} f(y) dy $$

$$
\begin{aligned}
&= \frac{1}{\Gamma(p)} \int_c^x (x-y)^{p-1} \frac{1}{\Gamma(q)} \int_c^y (y-z)^{q-1} f(z) dz dy \\
&= \frac{1}{\Gamma(p)} \frac{1}{\Gamma(q)} \int_c^x \int_c^y (x-y)^{p-1} (y-z)^{q-1} f(z) dz dy \\
&= \frac{1}{\Gamma(p)} \frac{1}{\Gamma(q)} \int_c^x \int_z^x (x-y)^{p-1} (y-z)^{q-1} dy f(z) dz,
\end{aligned}
$$

we perform the substitution $y = (x-z)w + z$ then,

$$
\begin{aligned}
&= \frac{1}{\Gamma(p)} \frac{1}{\Gamma(q)} \int_c^x \int_0^1 (x-z)^{p+q-1} (1-w)^{p-1} w^{q-1} dw f(z) dz \\
&= \frac{1}{\Gamma(p)} \frac{1}{\Gamma(q)} \int_c^x (x-z)^{p+q-1} B(p,q) f(z) dz \\
&= \frac{1}{\Gamma(p+q)} \int_c^x (x-z)^{p+q-1} f(z) dz \\
&= {}_c\mathbf{D}_x^{-p-q} f(x)
\end{aligned}
$$

where $B(p,q)$ is the beta function and we used the relation $B(p,q) = \Gamma(p)\Gamma(q)/\Gamma(p+q)$ [3]. □

Also we get the commutativity law as a byproduct:

$$
{}_c\mathbf{D}_x^{-p} \left({}_c\mathbf{D}_x^{-q} f(x) \right) = {}_c\mathbf{D}_x^{-p-q} f(x) = {}_c\mathbf{D}_x^{-q} \left({}_c\mathbf{D}_x^{-p} f(x) \right).
$$

Now we extended integer order integral to real order. We apply derivative to real order integral to extend integer order derivative. For a real $\alpha > 0$, we take integer $k > 0$ and define the following:

$$
{}_c\mathbf{D}_x^{k-\alpha} f(x) = \frac{1}{\Gamma(\alpha)} \frac{d^k}{dt^k} \int_c^x (x-y)^{\alpha-1} f(y) dy \tag{3.19}
$$

Actually, as we defined in (3.16), the only restriction for α in (3.19) is $\alpha > 0$. And we can replace the condition $\alpha > 0$ with the narrower condition $0 < \alpha \leq 1$. By denoting $p = k - \alpha$ in (3.19), we define Rienamm-Liouville fractional derivative.

Definition 3.1.6. *For an positive real p, Riemann-Liouville(RL) fractional derivative is defined as follows:*

$$ {}_c\mathbf{D}_x^p f(x) =: \frac{1}{\Gamma(k-p)} \frac{d^k}{dx^k} \int_c^x (x-y)^{k-p-1} f(y) dy, \qquad (k-1 \le p < k) $$

Note that if p is an integer, then RL fractional derivative is a conventional integer-order derivative.

Now we present some properties of the RL fractional derivatives.

Proposition 3.1.7. *For $p > 0, q > 0$ and $x > c$, RL fractional derivatives of continuous function $f(x)$ satisfies the following equation*

$$ i) \quad {}_c\mathbf{D}_x^p \left({}_c\mathbf{D}_x^{-p} f(x) \right) = f(x) \tag{3.20} $$

And assume if $p \ge q \ge 0$, then ${}_c\mathbf{D}_x^{p-q} f(x)$ exist. Then

$$ ii) \quad {}_c\mathbf{D}_x^p \left({}_c\mathbf{D}_x^{-q} f(x) \right) = {}_c\mathbf{D}_x^{p-q} f(x) \tag{3.21} $$

Proof. i) When p is an integer, the RL fractional derivative is a conventional derivative, so the equation (3.20) is trivially satisfied. Let us consider the case $k-1 \le p < k$ for the integer k. By the proposition 3.1.5,

$$ \begin{aligned} {}_c\mathbf{D}_x^p \left({}_c\mathbf{D}_x^{-p} f(x) \right) &= {}_c\mathbf{D}_x^k \left({}_c\mathbf{D}_x^{-(k-p)} \left({}_c\mathbf{D}_x^{-p} f(x) \right) \right) \\ &=_c \mathbf{D}_x^k \left({}_c\mathbf{D}_x^{-k} f(x) \right) = f(x). \end{aligned} $$

ii) If $q \ge p \ge 0$, then

$$ \begin{aligned} {}_c\mathbf{D}_x^p \left({}_c\mathbf{D}_x^{-q} f(x) \right) &= {}_c\mathbf{D}_x^p \left({}_c\mathbf{D}_x^{-p} \left({}_c\mathbf{D}_x^{-(q-p)} f(x) \right) \right) \\ &= {}_c\mathbf{D}_x^{-(q-p)} f(x) = {}_c\mathbf{D}_x^{p-q} f(x) \end{aligned} $$

Now let us assume $p > q \ge 0$. Define integers m and n such that $0 \le m-1 \le p < m$ and $0 \le n-1 \le p-q < n$. Then $n \le m$ and

$$ {}_c\mathbf{D}_x^p \left({}_c\mathbf{D}_x^{-q} f(x) \right) = {}_c\mathbf{D}_x^m \left({}_c\mathbf{D}_x^{-(m-p)} \left({}_c\mathbf{D}_x^{-q} f(x) \right) \right) $$

$$
\begin{aligned}
&= {}_c\mathbf{D}_x^m \left({}_c\mathbf{D}_x^{-(m-p+q)} f(x)\right) \\
&= {}_c\mathbf{D}_x^n \left({}_c\mathbf{D}_x^{-(n-(p-q))} f(x)\right) = {}_c\mathbf{D}_x^{p-q} f(x)
\end{aligned}
$$

□

Next proposition shows the non-commutativity of integral and differential of RL.

Proposition 3.1.8. *Suppose ${}_c\mathbf{D}_x^p f(x)$ is integrable and $k-1 \leq p < k$ for an integer k. Then the following equations holds:*

$$
i) \quad {}_c\mathbf{D}_x^{-p} \left({}_c\mathbf{D}_x^p f(x)\right) = f(x) - \sum_{i=1}^{k} \left({}_c\mathbf{D}_x^{p-i} f(x)\right)_{x=c} \frac{(x-c)^{p-i}}{\Gamma(p-i+1)}.
$$

$$
ii) \quad {}_c\mathbf{D}_x^{-p} \left({}_c\mathbf{D}_x^q f(x)\right) = {}_c\mathbf{D}_x^{q-p} f(x) - \sum_{i=1}^{k} \left({}_c\mathbf{D}_x^{p-i} f(x)\right)_{x=c} \frac{(x-c)^{p-i}}{\Gamma(p-i+1)}.
$$

Proof. i) Let us first observe the following:

$$
\begin{aligned}
\frac{1}{\Gamma(p+1)} \int_c^x (x-y)^p \ {}_c\mathbf{D}_y^p f(y) dy &= \frac{1}{\Gamma(p+1)} \int_c^x (x-y)^p \frac{d^k}{dy^k} \left({}_c\mathbf{D}_y^{-(k-p)} f(y)\right) dy \\
&= \frac{1}{\Gamma(p)} \int_c^x (x-y)^{p-1} \frac{d^{k-1}}{dy^{k-1}} \left({}_c\mathbf{D}_y^{-(k-p)} f(y)\right) dy \\
&\quad - \frac{1}{\Gamma(p+1)} (x-y)^p \ \frac{d^{k-1}}{dy^{k-1}} \left({}_c\mathbf{D}_y^{-(k-p)} f(y)\right)\Big|_{y=c} \\
&= \dots \\
&= \frac{1}{\Gamma(p-k+1)} \int_c^x (x-y)^{p-k} \ {}_c\mathbf{D}_y^{-(k-p)} f(y) dy \\
&\quad - \sum_{i=1}^{k} \frac{1}{\Gamma(p+2-i)} (x-y)^{p-i+1} \frac{d^{k-i}}{dy^{k-i}} \left({}_c\mathbf{D}_y^{-(k-p)} f(y)\right)_{y=c} \\
&= {}_c\mathbf{D}_x^{-(p-k+1)} \left({}_c\mathbf{D}_x^{-(k-p)} f(x)\right) - \sum_{i=1}^{k} \frac{(x-y)^{p-i+1}}{\Gamma(p-i+2)} \left({}_c\mathbf{D}_y^{p-i} f(y)\right)_{y=c}
\end{aligned}
\tag{3.22}
$$

Because of the integrablity of ${}_c\mathbf{D}_x^p f(x)$, the fractional derivatives ${}_c\mathbf{D}_x^{p-i} f(x)$ $(i = 1, \ldots, k)$ are all bounded at $x = c$ and the inetgral values in (3.22) exist. And we used Leibnitz' rule and integration by parts k times in (3.22). Then,

$$\begin{aligned}
{}_c\mathbf{D}_x^{-p}\left({}_c\mathbf{D}_x^p f(x)\right) &= \frac{1}{\Gamma(p)} \int_c^x (x-y)^{p-1}\ {}_c\mathbf{D}_y^p f(y) dy \\
&= \frac{d}{dx}\left(\frac{1}{\Gamma(p+1)} \int_c^x (x-y)^p\ {}_c\mathbf{D}_y^p f(y) dy \right) \\
&= \frac{d}{dx}\left({}_c\mathbf{D}_x^{-1} f(x) - \sum_{i=1}^{k} \frac{(x-y)^{p-i+1}}{\Gamma(p-i+2)} \left({}_c\mathbf{D}_y^{p-i} f(y) \right)_{y=c} \right) \\
&= f(x) - \sum_{i=1}^{k} \frac{(x-c)^{p-i}}{\Gamma(p-i+1)} \left(\ {}_c\mathbf{D}_x^{p-i} f(x) \right)_{x=c}
\end{aligned}$$

ii) To prove this equation, we need the RL fractional derivative formula for the power function. For those interested in this proof, see [41]. □

3.2 Proposed model: Cauchy noise removal model by fractional total variation

Last section, we defined some fractional derivatives. In this section, we define fractional total variation by using fractional derivatives and study the properties of the fractional total variation (TV) and propose the fractional TV model for Cauchy noise removal. To solve the proposed model, we introduce nonconvex alternating direction method of multipliers (nonconvex ADMM). Finally, we present algorithm solving the proposed model by using nonconvex ADMM and proximal operator. For convenience, we assume a fractional order α is lie in $n-1 < \alpha < n$ for a positive integer n and a bounded domain Ω is a subset of $\mathbb{R}^d$ (mainly $d = 2$).

3.2.1 Fractional total variation and Cauchy noise removal model

Fractional total variation is generalization of conventional total variation. The solution of conventional TV lies in the function space BV(Ω). The same is true for fractional TV. To define fractional TV, we introduce the following definition. [59]

Definition 3.2.1. *Let us denote $\mathcal{C}^k(\Omega, \mathbb{R}^d)$ as the space of k-order continuously differentiable functions. And for any $\phi \in \mathcal{C}^k(\Omega, \mathbb{R}^d)$, assume the $(k+1)$-th order derivative $\phi^{(k+1)}$ is integrable and $\dfrac{\partial^i \phi(x)}{\partial n^i}|_{\partial\Omega} = 0$ for all $i = 0, 1, \ldots, k$, then ϕ is a compactly supported continuous integrable function in Ω. And we denote $\mathcal{C}_0^k(\Omega, \mathbb{R}^d)$ as the k-compackly supported continuous-integrable function space.*

By using the space of the test functions, we can define fractional α-order total variation.

Definition 3.2.2. *The α-order total variation of u is defined by*

$$\mathrm{TV}^\alpha(u) := \sup_{\phi \in \mathcal{S}} \int_\Omega (-u \ \mathrm{div}^\alpha \phi)\, dx \tag{3.23}$$

where $\mathcal{S}$ be the domain of some test functions

$$\mathcal{S} := \left\{ \phi \in \mathcal{C}_0^k(\Omega, \mathbb{R}^d) \middle| |\phi(x)| \leq 1 \text{ for all } x \in \Omega \right\},$$

and $|\phi| = \sqrt{\sum_{i=1}^d \phi_i^2}$.

The above fractional divergence is similar with the ordinary divergence. Concretely, $\mathrm{div}^\alpha \phi = \sum_{i=1}^d \dfrac{\partial^\alpha \phi_i}{\partial x_i^\alpha}$ and $\dfrac{\partial^\alpha \phi_i}{\partial x_i^\alpha} = {}_a D_b^\alpha \phi_i$ (fractional derivative of ϕ_i along the x_i direction). Also, if we use $TV^\alpha(u)$, then we can define the space of functions of α-bounded variation on Ω:

$$\mathrm{BV}^\alpha(\Omega) := \left\{ u \in \mathrm{L}^1(\Omega) \middle| \mathrm{TV}^\alpha(u) < +\infty \right\}.$$

Now we can present our first model for Cauchy noise denoising. In section 2.2, we derive the variational model by using MAP and Gibbs prior. Now we apply fractional TV to J in (2.12). Then we get the following:

$$\arg\min_u \mathrm{TV}^\alpha(u) + \frac{\lambda}{2}\int_\Omega (\log(\gamma^2 + (u(x) - f(x))^2)dx \tag{3.24}$$

To slove (3.24), we first need to transform the TV^α into a dual form that is easier to handle. To do this we follow Zhang and Chen's work [59]. For any positive integer p, let $W_p^\alpha(\Omega) = \left\{u \in L^p(\Omega) \middle| ||u||_{W_p^\alpha(\Omega)} < +\infty\right\}$ be a function space with the norm

$$||u||_{W_p^\alpha(\Omega)} = \left(\int_\Omega |u|^p dx + \int_\Omega |\nabla^\alpha u|^p dx\right)^{1/p}, \qquad \nabla^\alpha u = \left(\frac{\partial^\alpha u}{\partial x_1}, \cdots, \frac{\partial^\alpha u}{\partial x_d}\right)^T.$$

According to the [2], the α-order integration by parts is satisfied as follows:

$$\begin{aligned}
&\int_a^b f(x)\ {}_aD_x^\alpha g(x)dx \\
&= (-1)^n \int_a^b {}_aD_x^\alpha f(x)\cdot g(x)dx + \sum_{i=0}^{n-1}(-1)^i\ {}_aD_x^{\alpha-n+i} f(x)\cdot \frac{d^{n-i-1}g(x)}{dx^{n-i-1}}\bigg|_{x=a}^b \\
&= (-1)^n \int_a^b {}_aD_x^\alpha f(x)\cdot g(x)dx
\end{aligned}$$

for $f \in W_1^\alpha([a,b]), g \in \mathcal{C}_0^{n-1}([a,b],\mathbb{R})$ and in the second equation, we used the compact supportness of g. And for the multivariable case (our case),

$$\int_\Omega u(x)\left((-1)^n \mathrm{div}^\alpha\right)\phi(x)dx = \int_\Omega \nabla^\alpha u(x)\cdot\phi(x)dx, \tag{3.25}$$

where $u \in W_1^\alpha(\Omega), \phi \in \mathcal{C}_0^{n-1}(\Omega,\mathbb{R}^d)$.

Proposition 3.2.3. *If* $u \in W_1^\alpha(\Omega)$, *then* $\mathrm{TV}^\alpha(u) = \int_\Omega |\nabla^\alpha u(x)|dx$.

Proof. From (3.25),

$$\int_\Omega -u(x)\ \mathrm{div}^\alpha \phi(x)dx = (-1)^{n-1}\int_\Omega \nabla^\alpha u(x)\cdot\phi(x)dx \tag{3.26}$$

and if we take

$$\phi_0 = \begin{cases} (-1)^{n-1}\dfrac{\nabla^\alpha u}{|\nabla^\alpha u|} & if\ |\nabla^\alpha u(x)| \neq 0 \\ 0 & otherwise \end{cases}$$

then $|\phi_0| \leq 1$ and it maximize the functional $\int_\Omega \phi(x)\cdot\nabla^\alpha u(x)dx = \int_\Omega |\nabla^\alpha u(x)|dx$. Introduce mollifying $\eta_\epsilon \in \mathcal{C}_0^{n-1}(\Omega, \mathbb{R}^d)$ and define mollified function $\phi_\epsilon = \phi_0 * \eta_\epsilon$ where $*$ is the convolution. Then $\phi_\epsilon \in S$ and

$$\lim_{\epsilon\to 0}\int_\Omega \phi_\epsilon \cdot \nabla^\alpha u(x)dx = \int_\Omega |\nabla^\alpha u(x)|dx$$

Therefore $\mathrm{TV}^\alpha(u) = \sup_{\phi\in\mathcal{S}}\int_\Omega (-u\ \mathrm{div}^\alpha\phi)\,dx = \int_\Omega |\nabla^\alpha u(x)|dx$. □

Therefore, (3.24) can be changed the following form:

$$\arg\min_u \int_\Omega |\nabla^\alpha u(x)|dx + \frac{\lambda}{2}\int_\Omega (\log(\gamma^2 + (u(x) - f(x))^2)dx \tag{3.27}$$

Similar to TV, we can show that fractional TV is convex by using linearity of fractional-order derivatives, the positively homogeneous and subadditive properties of fractional TV. But due to the data fidelity term of (3.27), the model (3.27) is nonconvex that makes hard to solve the minimization problem. To solve this, we introduce the recent result in [52] which is nonconvex ADMM algorithm.

3.2.2 nonconvex ADMM algorithm

Let us consider the following optimization problem:

$$\min_{\mathbf{u},v} \mathcal{F}(\mathbf{u}) + \mathcal{G}(v)$$
$$\text{subject to} \quad \mathbf{A}\mathbf{u} + Bv = 0$$

where $\mathcal{F}$ is a continuous and $\mathcal{G}$ is a differentiable function, and a vector $\mathbf{u} = [u_1; \cdots; u_p] \in \mathbb{R}^N,\ u_i \in \mathbb{R}^{n_i}$ is a variable with corresponding coefficient $\mathbf{A} = [A_1, \cdots, A_p] \in \mathbb{R}^{M\times N},\ A_i \in \mathbb{R}^{M\times n_i}, \sum_{i=1}^{p} n_i = N$ and $B \in \mathbb{R}^{M\times L}$ is a linear operator of $v \in \mathbb{R}^L$. Alternating direction method of multipliers says that the algorithm 1 (below) converges if $\mathcal{F}$ and $\mathcal{G}$ are convex functions. Also Wang et al. proved that if some conditions are satisfied, then the algorithm 1 converges even though $\mathcal{F}$ is nonsmooth and nonconvex, and $\mathcal{G}$ is nonconvex. We form the augmented Lagrangian by introducing a Lagrangian multiplier $w \in \mathbb{R}^M$ of the constraint $\mathbf{A}\mathbf{u} + Bv = 0$:

$$\mathcal{L}_\rho(\mathbf{u}, v, w) = \mathcal{F}(\mathbf{u}) + \mathcal{G}(v) + \langle w, \mathbf{A}\mathbf{u} + Bv\rangle + \frac{\rho}{2}\|\mathbf{A}\mathbf{u} + Bv\|_2^2 \tag{3.28}$$

where $\rho > 0$ is a penalty parameter and $\langle\cdot,\cdot\rangle$ is the inner product. And the (nonconvex) ADMM algorithm is

Algorithm 1 ADMM algorithm

Inputs $u_1^0, \ldots, u_p^0, v^0, w^0$.
while the convergence condition is not satisfied **do**
  for $i = 1, \ldots, p$ **do**
    $u_i^{k+1} = \arg\min_{u_i} \mathcal{L}_\rho(u_{<i}^{k+1}, u_i, u_{>i}^{k}, v^k, w^k)$
  end for
  $v^{k+1} = \arg\min_{v} \mathcal{L}_\rho(\mathbf{u}^{k+1}, v, w^k)$
  $w^{k+1} = w^k + \rho(\mathbf{A}\mathbf{u}^{k+1} + Bv^{k+1})$
  $k = k + 1$
end while.

Before give the convergence theorem, we need some definitions.

Definition 3.2.4. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called piecewise linear function if there exist disjoint sets $U_1, \ldots, U_K \subset \mathbb{R}^n$ (i.e. $U_i \cap U_j = \emptyset$ $(i \neq j)$) and vectors $a_1, \ldots, a_K \in \mathbb{R}^n$ and $b_1, \ldots, b_K \in \mathbb{R}$ such that $\cup_{i=1}^K \overline{U}_i = \mathbb{R}^n$, and $f(x) = a_i^T x + b_i$ for each set U_i $(i = 1, \ldots, K)$.*

Definition 3.2.5. *A lower semi-continuous function $f : \mathbb{R}^N \to \mathbb{R}$ is called restricted prox-regular if, for any $M > 0$ and bounded set $T \subset dom(f)$ (where $dom(f)$ means the domain of a function f), there exist $c > 0$ such that*

$$f(y) + \frac{c}{2}\|x - y\|_2^2 \geq f(x) + \langle d, y - x\rangle$$

where

$$x \in T - S_M, y \in T, d \in \partial f(x), \|d\| \leq M$$
$$S_M = \{x \in dom(f) : \|d\| > M, d \in \partial f(x)\}.$$

Now we give the convergence conditions of ADMM algorithm under non-convexity and nonsmoothness in [52].

Theorem 3.2.6. *Suppose under the following assumptions are satisfied:*

1. *Define the set $\mathcal{S} := \{(\mathbf{u}, v) \in \mathbb{R}^{N+L} : \mathbf{A}\mathbf{u} + Bv = 0\}$. If $\|(\mathbf{u}, v)\| \to \infty$ for $(\mathbf{u}, v) \in \mathcal{F}$, then $\mathcal{F}(\mathbf{u}) + \mathcal{G}(v) \to \infty$ (In such case, $\mathcal{F}(\mathbf{u}) + \mathcal{G}(v)$ is called coercive over the $\mathcal{S}$).*

2. *$Im(\mathbf{A}) \subset Im(B)$, where $Im(\cdot)$ is the image of the given map.*

3. *(a) For any fixed $\mathbf{u}$ and c, $\arg\min_v\{\mathcal{F}(\mathbf{u}) + \mathcal{G}(v) : Bv = c\}$ has a unique minimizer. Then $H : Im(B) \to \mathbb{R}^L$ defined by $H(c) := \arg\min_v\{\mathcal{F}(\mathbf{u}) + \mathcal{G}(v) : Bv = c\}$ is a Lipschitz continuous map.*

   (b) For any fixed $u_{<i}, u_{>i}, v$ $(i = 1, \ldots, p)$ and c, $\arg\min_{u_i}\{\mathcal{F}(u_{<i}, u_i, u_{>i}) + \mathcal{G}(v) : A_i x_i = c\}$ has a unique minimizer. Then $J_i : Im(A_i) \to \mathbb{R}^{n_i}$ defined by $J_i(c) := \arg\min_{u_i}\{\mathcal{F}(u_{<i}, u_i, u_{>i}) + \mathcal{G}(v) : A_i x_i = c\}$ is a Lipschitz continuous map.

4. $\mathcal{F}$ *should has the following form*

$$\mathcal{F}(\mathbf{u}) = \sum_{i=1}^{p} f_i(u_i) + g(\mathbf{u})$$

and satisfy the followings

(a) $g(\mathbf{u})$ *is Lipshitz differentiable with constant* L_g*.*

(b) satisfy one of the followings

i. f_1 *is lower semi-continuous and* $f_i(u_i)$ *is restricted prox-regular for* $i = 2, \ldots, p$*.*

ii. $\sup\{\|d\| : u_0 \in S, d \in \partial f_1(u_1)\}$ *is bounend for any bounded set* S*,* $f_i(u_i)$ *is continuous and piecewise linear for* $i = 2, \ldots, p$*.*

5. $\mathcal{G}(v)$ *is Lipschitz differentiable with constant* $L_{\mathcal{G}}$*.*

Then for any sufficient large ρ *and any starting points* $\boldsymbol{u}^0, v^0, w^0$*, algorithm 1 generate a sequence that is bounded and that sequence has at least one limit point, and each limit point* $\boldsymbol{u}^*, v^*, w^*$ *is a stationary point of* $\mathcal{L}_\rho$ *(i.e.* $0 \in \mathcal{L}_\rho(\boldsymbol{u}^*, v^*, w^*)$*). Also if* $\mathcal{L}^\rho$ *satisfies the Kurdyka-Lojasiewicz (KL) inequality, then a stationary point turn to the unique global limit point.*

3.2.3 The algorithm for solving fractional total variational model of Cauchy noise

Now let us apply nonconvex ADMM to our variational model (3.27). The $\Omega = \mathbb{R}^{m\times n}$ is the domain of image, and $F : \Omega \to \mathbb{R}$ is the noisy image. The discrete form of our nonconvex model (3.27) is given as follows:

$$\arg\min_{U,V} \|\nabla^\alpha U\|_1 + \frac{\lambda}{2}\langle \log(\gamma^2 + (U - F)^2, \mathbf{1}\rangle. \tag{3.29}$$

where $\langle\cdot,\cdot\rangle$ is inner product for matirx (i.e. $\langle U, V\rangle = \sum_{i,j} U_{i,j}V_{i,j}$ for $U, V \in \mathbb{R}^{n\times m}$).

We introduce auxiliary variable $V \in \mathbb{R}^{n\times n}$ to construct augmented Lagrangian. Then (3.29) is rewritten as:

$$\arg\min_{U,V} \|\nabla^\alpha U\|_1 + \frac{\lambda}{2}\langle \log(\gamma^2 + (V-F)^2, \mathbf{1}\rangle \qquad s.t. \quad U = V. \tag{3.30}$$

We substitute $\|\nabla^\alpha U\|_1$ for $\mathcal{F}(U)$ and $\frac{\lambda}{2}\langle \log(\gamma^2 + (V-F)^2, \mathbf{1}\rangle$ for $\mathcal{G}(V)$ in the (3.28). Then the augmented Lagrangian of (3.30) is

$$\mathcal{L}_\rho(U,V,W) = \|\nabla^\alpha U\|_1 + \frac{\lambda}{2}\langle \log(\gamma^2 + (V-F)^2, \mathbf{1}\rangle + \langle W, U-V\rangle + \frac{\rho}{2}\|U-V\|_F^2.$$

where $\rho > 0$, $W \in \mathbb{R}^{n\times n}$ and $\|\cdot\|_F$ means the Frobenius norm. In the previous section, we present ADMM algorithm for multivariable $U_1, \ldots, U_p$, but, U is one variable in our problem. Therefore, Nonconvex ADMM algorithm has been simplified as follows:

Algorithm 2 ADMM algorithm for (3.29)

Inputs V^0, W^0.

while the convergence condition is not satisfied **do**

$U^{k+1} = \arg\min_U \|\nabla^\alpha U\|_1 + \frac{\rho}{2}\left\|U - V^k + \frac{W^k}{\rho}\right\|_F^2$

$V^{k+1} = \arg\min_V \frac{\lambda}{2}\langle \log(\gamma^2 + (V-F)^2, \mathbf{1}\rangle + \frac{\rho}{2}\left\|U^{k+1} - V + \frac{W^k}{\rho}\right\|_F^2$

$W^{k+1} = W^k + \rho(U^{k+1} - V^{k+1})$

$k = k + 1$

end while.

By giving some conditions of the fractional TV model (3.29), we can guarantee the convergence of algorithm 2.

Theorem 3.2.7. *Let (U^0, V^0, W^0) be any starting point and $\{(U^k, V^k, W^k)\}$ be generated sequence points by algorithm 2. If $\rho > \frac{\lambda}{\gamma^2}$, the $\{(U^k, V^k, W^k)\}$ converges globally to a stationary point (U^*, V^*, W^*) of $\mathcal{L}_\rho$.*

To prove the theorem 3.2.7, we will use some lemmas. Let us define the

following:

$$\mathcal{F} : \mathbb{R}^{n\times n} \to \mathbb{R}, \quad \mathcal{F}(U) = \|\nabla^\alpha U\|_1.$$
$$\mathcal{G} : \mathbb{R}^{n\times n} \to \mathbb{R}, \quad \mathcal{G}(V) = \frac{\lambda}{2}\langle \log(\gamma^2 + (V-F)^2), \mathbf{1}\rangle.$$

Lemma 3.2.8. *The iterates $\{(U^k, V^k, W^k)\}_{k\in\mathbb{N}}$ in Algorithm 2 satisfies:*

1. *$\mathcal{L}_\rho(U^k, V^k, W^k)$ is lower-bounded and non-increasing for all $k \in \mathbb{N}$*

2. *$\{(U^k, V^k, W^k)\}$ is bounded.*

Proof. By the definition of U^{k+1}, we can obtain

$$\mathcal{L}_\rho(U^k, V^k, W^k) - \mathcal{L}_\rho(U^{k+1}, V^k, W^k) \geq 0.$$

Furthermore, because

$$\begin{aligned}
&\mathcal{L}_\rho(U^{k+1}, V^k, W^k) - \mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^k)\\
&= \mathcal{G}(V^k) - \mathcal{G}(V^{k+1}) - \langle W^k, V^k - V^{k+1}\rangle\\
&- \rho\langle U^{k+1} - V^{k+1}, V^k - V^{k+1}\rangle + \frac{\rho}{2}\|V^k - V^{k+1}\|^2\\
&= \mathcal{G}(V^k) - \mathcal{G}(V^{k+1}) - \langle W^{k+1}, V^k - V^{k+1}\rangle + \frac{\rho}{2}\|V^k - V^{k+1}\|^2,
\end{aligned}$$

$$\begin{aligned}
&\mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^k) - \mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1})\\
&= \langle W^k - W^{k+1}, U^{k+1} - V^{k+1}\rangle = -\frac{1}{\rho}\|W^k - W^{k+1}\|^2,
\end{aligned}$$

we have the following

$$\begin{aligned}
&\mathcal{L}_\rho(U^{k+1}, V^k, W^k) - \mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1})\\
&\quad= \mathcal{G}(V^k) - \mathcal{G}(V^{k+1}) - \langle W^{k+1}, V^k - V^{k+1}\rangle\\
&\quad- \frac{1}{\rho}\|W^k - W^{k+1}\|^2 + \frac{\rho}{2}\|V^k - V^{k+1}\|^2. \qquad (3.31)
\end{aligned}$$

Here, from the first-order optimality condition of line 4 in Algorithm 2

$$\nabla\mathcal{G}(V^k) - W^{k-1} + \rho(V^k - U^k) = 0,$$

so we have,

$$\nabla\mathcal{G}(V^k) = W^k, \tag{3.32}$$

and the smoothness of $\mathcal{G}$ implies

$$\|W^{k+1} - W^k\| = \|\nabla\mathcal{G}(V^{k+1}) - \nabla\mathcal{G}(V^k)\| \leq L_{\nabla\mathcal{G}}\|V^{k+1} - V^k\|, \tag{3.33}$$

where the Lipschitz constant $L_{\nabla\mathcal{G}} = \dfrac{\lambda}{\gamma^2}$ follows from $\nabla^2\mathcal{G} \leq \dfrac{\lambda}{\gamma^2}$.
Then (3.31) becomes

$$\begin{aligned}
&\mathcal{G}(V^k) - \mathcal{G}(V^{k+1}) - \langle\nabla\mathcal{G}(V^{k+1}), V^k - V^{k+1}\rangle \\
&- \frac{1}{\rho}\|W^k - W^{k+1}\|^2 + \frac{\rho}{2}\|V^k - V^{k+1}\|^2 \\
&\quad \geq \left(-\frac{L_{\nabla\mathcal{G}}}{2} - \frac{L_{\nabla\mathcal{G}}^2}{\rho} + \frac{\rho}{2}\right)\|V^k - V^{k+1}\|^2.
\end{aligned}$$

from (3.33) and from the fact that $\nabla\mathcal{G}$ is Lipschitz continuous.
If we let $C_1 = -\dfrac{L_{\nabla\mathcal{G}}}{2} - \dfrac{L_{\nabla\mathcal{G}}^2}{\rho} + \dfrac{\rho}{2} > 0$ which is equivalent to $\rho > \dfrac{2\lambda}{\gamma^2}$, we have

$$\mathcal{L}_\rho(U^k, V^k, W^k) - \mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1}) \geq C_1\|V^k - V^{k+1}\|^2, \tag{3.34}$$

indicating that $\mathcal{L}_\rho(U^k, V^k, W^k)$ is non-increasing for $k \in \mathbb{N}$.

Now considering that $\nabla\mathcal{G}$ is Lipschitz continuous,

$$\begin{aligned}
&\mathcal{F}(U^k) \geq 0, \quad \mathcal{G}(V^k) \geq \frac{\lambda}{2}\langle\log\gamma^2, \mathbf{1}\rangle, \\
&\mathcal{G}(V^k) + \langle W^k, U^k - V^k\rangle + \frac{\rho}{2}\|U^k - V^k\|^2 \\
&\qquad = \mathcal{G}(V^k) + \langle\nabla\mathcal{G}(V^k), U^k - V^k\rangle + \frac{\rho}{2}\|U^k - V^k\|^2
\end{aligned}$$

$$\begin{aligned}
&\geq \mathcal{G}(U^k) - \frac{L_{\nabla\mathcal{G}}}{2}\|U^k - V^k\|^2 + \frac{\rho}{2}\|U^k - V^k\|^2 \\
&\geq \frac{\lambda}{2}\langle \log\gamma^2, \mathbf{1}\rangle.
\end{aligned}$$

Thus, $\mathcal{L}_\rho(U^k, V^k, W^k)$ is lower-bounded. Because $\mathcal{L}_\rho(U^k, V^k, W^k)$ is non-increasing and $\mathcal{F}$ is coercive, $\{(U^k, V^k)\}$ is bounded. Additionally, from (3.32), it is clear that $\{W^k\}$ is bounded. □

Lemma 3.2.9. *For all $k \geq 1$, there exists a constant $C_2 > 0$ and $p^{k+1} \in \partial\mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1})$ such that $\|p^{k+1}\| \leq C_2\|V^k - V^{k+1}\|$.*

Proof. Here, we denote the partial of Lagrangian $\partial\mathcal{L}_\rho$ by

$$\partial\mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1}) = (\partial_U\mathcal{L}_\rho, \nabla_V\mathcal{L}_\rho, \nabla_W\mathcal{L}_\rho)(U^{k+1}, V^{k+1}, W^{k+1}).$$

By direct computation and (3.33),

$$\begin{aligned}
&\|\nabla_V\mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1})\| = \|W^k - W^{k+1}\| \leq L_{\mathcal{G}}\|V^k - V^{k+1}\|. \\
&\|\nabla_W\mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1})\| = \frac{1}{\rho}\|W^k - W^{k+1}\| \leq \frac{L_{\mathcal{G}}}{\rho}\|V^k - V^{k+1}\|.
\end{aligned}$$

Furthermore, observe that

$$\begin{aligned}
\partial_U\mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1}) &= \partial_U\mathcal{F}(U^{k+1}) + W^{k+1} + \rho(U^{k+1} - V^{k+1}) \\
&= \partial_U\mathcal{F}(U^{k+1}) + W^k + \rho(U^{k+1} - V^k) \\
&\quad + (W^{k+1} - W^k) + \rho(V^k - V^{k+1}).
\end{aligned}$$

From the optimality condition of line 4 in Algorithm 2, we have $0 \in \partial_U\mathcal{F}(U^{k+1}) + W^k + \rho(U^{k+1} - V^k)$. Therefore if we define p^{k+1} as follows:

$$p^{k+1} := \left((W^{k+1} - W^k) + \rho(V^k - V^{k+1}),\ W^k - W^{k+1},\ \frac{1}{\rho}(W^k - W^{k+1})\right),$$

then $p^{k+1} \in \partial\mathcal{L}_\rho(U^{k+1}, V^{k+1}, W^{k+1})$ and

$$\|p^{k+1}\| \leq \left(L_{\mathcal{G}}(2 + \frac{1}{\rho}) + \rho\right)\|V^k - V^{k+1}\|,$$

where $C_2 = L_{\mathcal{G}}(2 + \frac{1}{\rho}) + \rho$. □

(Proof of Theorem 3.2.7). Because $\{(U^k, V^k, W^k)\}$ is bounded, there exists a subsequence $\{(U^{k_s}, V^{k_s}, W^{k_s})\}$ which converges to (U^*, V^*, W^*) as $s \to \infty$. Because $\mathcal{L}_\rho(U^k, V^k, W^k)$ is non-increasing and lower-bounded, it converges and $\|V^k - V^{k+1}\| \to 0$ as $k \to \infty$ according to (3.34). From Lemma 3.2.9, there exists a sequence of subdifferentials $p^k \in \partial\mathcal{L}_\rho(U^k, V^k, W^k)$ that satisfies $\|p^k\| \to 0$ as $k \to \infty$. In particular, $\|p^{k_s}\| \to 0$ as $s \to \infty$. Continuity of $\mathcal{F}$ implies the continuity of $\mathcal{L}_\rho$ and $\lim_{s\to\infty} \mathcal{L}_\rho(U^{k_s}, V^{k_s}, W^{k_s}) = \mathcal{L}_\rho(U^*, V^*, W^*)$. Consequently, we have $0 \in \partial\mathcal{L}_\rho(U^*, V^*, W^*)$. □

In the algorithm 2, u-subproblem and v-subproblem are dominant steps. v-subproblem can be solved by Newton method, because of the property of twice continuously differentiable. The following algorithm solves the v-subproblem.

Algorithm 3 Algorithm for v-subproblem in Algorithm 2

Set $N \in \mathbb{N}$ and **Input** V_0, U^{k+1}, W^k, F.
for n = 0 to N-1 **do**

$$M = \frac{\lambda(V_n - F)}{\gamma^2 + (V_n - F)^2} - \rho\left(U^k - V_n + \frac{W^k}{\rho}\right)$$

$$D = \frac{\lambda(\gamma^2 - (V_n - F)^2)}{(\gamma^2 + (V_n - F)^2)^2} + \rho$$

$$V_{n+1} = V_n - \frac{M}{D}$$

end for
Return V_N as V^{k+1}

As we stated before, fractional TV operator is convex, so, u-subproblem is convex. To attack the u-subproblem, we apply proximity algorithm [10] and split-Bregman algorithm [19].
Before get into solving the u-subproblem, we specify which frcational derivative of this model is. For a vector $f = (f_0, f_1, \ldots, f_{n-1})^T \in \mathbb{R}^n$, the discretization of the left RL fractional is as follows [42, 41]:

$$\nabla^\alpha f = (f_0^\alpha, f_1^\alpha, \cdots, f_{n-1}^\alpha)^T = \hat{B}_n^\alpha (f_0, f_1, \cdots, f_{n-1})^T \tag{3.35}$$

where B_n^α is the upper triangular strip matrix:

$$\hat{B}_n^\alpha = \begin{pmatrix} \omega_0^\alpha & \omega_1^\alpha & \ddots & \ddots & \omega_{n-1}^\alpha \\ 0 & \omega_0^\alpha & \omega_1^\alpha & \ddots & \omega_{n-2}^\alpha \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \omega_0^\alpha & \omega_1^\alpha \\ 0 & 0 & \cdots & 0 & \omega_0^\alpha \end{pmatrix}$$

and ω^α is obtained recursively from

$$\omega_0^\alpha = 1, \qquad \omega_i^\alpha = (-1)^i \left(1 - \frac{\alpha+1}{i}\right) \omega_{i-1}^\alpha, \quad i = 1, 2, \ldots, n-1.$$

Similarly, the right sided RL derivative of $f = (f_0, f_1, \ldots, f_{n-1})^T \in \mathbb{R}^n$ is:

$$(f_0^\alpha, f_1^\alpha, \cdots, f_{n-1}^\alpha)^T = \hat{R}_n^\alpha (f_0, f_1, \cdots, f_{n-1})^T \tag{3.36}$$

where $\hat{R}_n^\alpha$ is the lower triangular strip matrix:

$$\hat{R}_n^\alpha = \begin{pmatrix} \omega_0^\alpha & 0 & \cdots & \cdots & 0 \\ \omega_1^\alpha & \omega_0^\alpha & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_{n-2}^\alpha & \ddots & \omega_1^\alpha & \omega_0^\alpha & 0 \\ \omega_{n-1}^\alpha & \omega_{n-2}^\alpha & \ddots & \omega_1^\alpha & \omega_0^\alpha \end{pmatrix}$$

And according to the Podlubny [41], the discretized left and right GL fractional derivative definition is same with (3.35) and (3.36), respectively. Instead left or right fractional derivative, we use a central fractional derivative that is not biased left or right:

$$B_n^\alpha = \hat{B}_n^\alpha + \hat{R}_n^\alpha. \tag{3.37}$$

We apply discretization of the fractional derivative to the two dimensional case. The discretized (left) RL fractional derivative of (i, j) component of

$U \in \mathbb{R}^{m\times n}$ is obtained by

$$\nabla_x^\alpha U_{i,j} = (B_n^\alpha U_{*,j})_i, \qquad \nabla_y^\alpha U_{i,j} = (U_{i,*}(B_n^\alpha)^T)_j$$

where $U_{*,j}$ means j-th column vector of U and $u_{i,*}$ means i-th row vector of U.

Similar with usual TV model, fractional TV model has two cases which are the anisotropic and isotropic model. Therefore, we consider the following two cases of u-subproblem in the algorithm 2:

$$\textbf{Anisotropic case :}\ \min_U \sum_{i,j} \left(|\nabla_x^\alpha U_{i,j}| + |\nabla_y^\alpha U_{i,j}|\right) + \frac{\rho}{2}\|U-\eta\|_2^2 \quad (3.38)$$

$$\textbf{Isotropic case :}\ \min_U \sum_{i,j} \sqrt{(\nabla_x^\alpha U_{i,j})^2 + (\nabla_y^\alpha U_{i,j})^2} + \frac{\rho}{2}\|U-\eta\|_2^2. \quad (3.39)$$

For computational convenience, we use the substitution $V^k - W^k/\rho = \eta$.

As we stated before, we will solve u-subproblem using both the proximity algorithm and the split Bregman iteration. Fortunately, both anisotropic and isotropic case can be solved by the proximity algorithm or the split Bregman iteration.

First, we solve the u-subproblem by the proximity algorithm. The proximity algorithm is easy to use and gives a convenient convergence for fractional TV denoising of Gaussian noise model. Proximity operator studied by Micchelli et al. [36] is given as follows.

Definition 3.2.10. *Let $\Phi : \mathbb{R}^N \to \mathbb{R}$ is convex function. Then the proximity operator of Φ at $p \in \mathbb{R}^N$ is*

$$\mathbf{prox}_\Phi p = \arg\min_{q\in\mathbb{R}^N} \left\{ \Phi(q) + \frac{1}{2}\|p-q\|_2^2 \right\} \quad (3.40)$$

Intuitively, proximity operator finds the point that is sufficiently close to the point p and minimize the value of Φ. For example, if $\Phi_1 = \frac{1}{\lambda}|\cdot|$, then

$$\mathbf{prox}_{\Phi_1} p = \max\left(|p| - 1/\lambda\right)\mathrm{sign}(p), \quad (3.41)$$

i.e. proximity operator is soft thersholding.

Since fractional TV is convex, we can apply proximity operator to (3.38). Consider $u \in \mathbb{R}^{n^2}$ is the vector expression of $U \in \mathbb{R}^{n\times n}$ and define $A^\alpha : \mathbb{R}^{n^2} \to \mathbb{R}^{2n^2}$. Such that $A^\alpha u$ is a vectorized expression of $(B_n^\alpha U, U(B_n^\alpha)^T)$.

Define $\Psi(u) := |\nabla_x^\alpha U| + |\nabla_y^\alpha U| = |\cdot| \circ A^\alpha u$ for anisotropic case and $\Psi(u) := \sqrt{(\nabla_x^\alpha U)^2 + (\nabla_y^\alpha U)^2} = |\cdot| \circ A^\alpha u$ for isotropic case. Then (3.38) is rewritten as

$$\arg\min_u \frac{1}{\rho}\Psi(u) + \frac{1}{2}\|u-\eta\|_2^2 = \mathbf{prox}_{\frac{1}{\rho}\Psi}\eta. \tag{3.42}$$

In order to solve the $\mathbf{prox}_{\frac{1}{\rho}\Psi}\eta$, we use the theorem in [36, 10].

Theorem 3.2.11. *Let $\Phi : \mathbb{R}^m \to \mathbb{R}$ is a convex function, M is an $m \times d$ matrix, $g \in \mathbb{R}^d$ and $\lambda > 0$. Define $B : \mathbb{R}^m \to \mathbb{R}^m$ is linear operator*

$$By := Mg + (I_m - \lambda MM^T)y \quad \textit{for} \quad y \in \mathbb{R}^m$$

and define $H : \mathbb{R}^m \to \mathbb{R}^m$ by

$$H := (I_m - \mathbf{prox}_{\frac{1}{\lambda}\Phi}) \circ B$$

then

$$\mathbf{prox}_{\Phi\circ M} g = g - \lambda M^T \omega$$

if and only if $\omega \in \mathbb{R}^m$ is a fixed point of H.

Theorem 3.2.11 says that finding a solution of proximity can be changed to finding the fixed point of some operator. Therefore, we can solve the $\mathbf{prox}_{\frac{1}{\rho}\Psi}\eta = \mathbf{prox}_{\frac{1}{\rho}|\cdot|\circ A^\alpha}\eta$ by finding the fixed point of $H = (I - \mathbf{prox}_{\frac{1}{\lambda\rho}|\cdot|}) \circ B$ where $By = A^\alpha\eta + (I - \lambda A^\alpha (A^\alpha)^T)y$ for a $y \in \mathbb{R}^{2n^2}$.

In short, if we apply the proximity algorithm to u-subproblem, we get the following algorithm:

Algorithm 4 proximity algorithm for (3.42)

Set error : $e > 0, \lambda > 0, 0 < \kappa < 1$.
Inputs $\omega_0 = 0, \rho$.
while $\dfrac{\|u^{n+1} - u^n\|_2}{\|u^n\|_2} > e$ **do**
$\tau^n = A^\alpha \eta + (I - \lambda A^\alpha (A^\alpha)^T)\omega^n$
$z^n = \mathbf{prox}_{\frac{1}{\lambda\rho}|\cdot|}\tau^n$
$\omega^{n+1} = \kappa\omega^n + (1-\kappa)(\tau^n - z^n)$
$u^{n+1} = \eta - \lambda(A^\alpha)^T\omega^{n+1}$
$n = n + 1$
end while.

where $\mathbf{prox}_{\frac{1}{\lambda\rho}|\cdot|}\tau^n$ is solved as (3.41) for anisotropic and solved by using a generalizd shrinkage formula [51] for isotropic case,

$$\mathbf{prox}_{\frac{1}{\lambda\rho}|\cdot|}\tau^n = \begin{cases} \operatorname{sign}(\tau^n)\max\left(|\tau^n| - \dfrac{1}{\lambda\rho}, 0\right) & \text{for anisotropic case} \\ \dfrac{(\tau_x^n, \tau_y^n)}{s^n}\max\left(s^n - \dfrac{1}{\lambda\rho}, 0\right) & \text{for isotropic case} \end{cases} \tag{3.43}$$

where $s^n = \sqrt{(\tau_x^n)^2 + (\tau_y^n)^2}$, $\tau^n = (\tau_x^n, \tau_y^n)$.

From now, we explained how to solve u-subproblem using the proximity algorithm. Next, we solve u-subproblem by using split Bregman iteration. The u-subproblem can be rewritten as follows:

$$\min_U \|\nabla^\alpha U\|_1 + \frac{\rho}{2}\|U - \eta\|_2^2.$$

To use split Bregman iteration, we consider the following constrained minimization problem:

$$\begin{aligned} \min_{U,D} |(D_x, D_y)| + \frac{\rho}{2}\|U - \eta\|_2^2 + \frac{\mu}{2}\|D - \nabla^\alpha U\|_2^2 \\ \text{s.t.} \quad D = (D_x, D_y) = (\nabla_x^\alpha U, \nabla_y^\alpha U) \end{aligned} \tag{3.44}$$

where $|(D_x, D_y)| = \sum_{i,j}(|D_{x,i,j}| + |D_{y,i,j}|)$ for anisotropic case and $|(D_x, D_y)| = \sum_{i,j}\sqrt{(D_{x,i,j})^2 + (D_{y,i,j})^2}$ for isotropic case, $D \in \mathbb{R}^{n\times 2n}$. In the above equa-

tion, We split L1 norm and L2 norm components of U. Then, to fit the constrained condition while solving the (3.44), we add Bregman term $R = (R_x, R_y) \in \mathbb{R}^{n\times 2n}$ which keep the constrained condition nearly by adding back the difference $D - \nabla^\alpha U$. In summary, we get the following two-phase iteration:

$$\begin{aligned}(U^{n+1}, D^{n+1}) &= \arg\min_{U,D} |(D_x, D_y)| + \frac{\rho}{2}\|U - \eta\|_2^2 + \frac{\mu}{2}\|D - \nabla^\alpha U - R^n\|_2^2 \\ R^{n+1} &= R^n + \nabla^\alpha U^{n+1} - D^{n+1}.\end{aligned}$$

Also, we perform efficiently by minimizing U and D alternately. Then we can write (U, D)-subproblem as follows:

$$\begin{aligned}U^{n+1} &= \arg\min_{U} \frac{\rho}{2}\|U - \eta\|_2^2 + \frac{\mu}{2}\|D^n - \nabla^\alpha U - R^n\|_2^2 \\ D^{n+1} &= \arg\min_{D} |(D_x, D_y)| + \frac{\mu}{2}\|D - \nabla^\alpha U^{n+1} - R^n\|_2^2\end{aligned} \tag{3.45}$$

Now U-subproblem and D-subproblem are much easier to solve. To emphasize linearity of fractional gradient, we use notation $B^\alpha U$ and $U(B^\alpha)^T$ instead of $\nabla_x^\alpha U$ and $\nabla_y^\alpha U$, respectively. Then we can get the solution by satisfying following optimal condition of U in (3.45):

$$\rho(U - \eta) - \mu(B^\alpha)^T(D_x^n - B^\alpha U - R_x^n) - \mu(D_y^n - U(B^\alpha)^T - R_y^n)B^\alpha = 0 \tag{3.46}$$

Except U-term, we pass the other terms to the right hand side and arrange the equation (3.46) as follows:

$$\begin{aligned}&\underbrace{\rho U + \mu(B^\alpha)^T B^\alpha U + \mu U(B^\alpha)^T B^\alpha}_{AU} \\ &\quad = \underbrace{\rho\eta + \mu(B^\alpha)^T D_x^n - \mu(B^\alpha)^T R_x^n + \mu D_y^n B^\alpha - \mu R_y^n B^\alpha}_{K}\end{aligned}$$

Then $AU = K$ is a linear problem. There are many ways to solve linear equations, in here, we use conjugate gradient method to solve $AU = K$. We have to check that A is a positive definite and symmetric to apply conjugate gradient. Because $(B^\alpha)^T B^\alpha$ is symmetric, A is also symmetric. We show

positive definite of A by following equations:

$$\begin{aligned}\langle AU, U\rangle &= \rho\langle U, U\rangle + \mu\langle (B^\alpha)^T B^\alpha U, U\rangle + \mu\langle U(B^\alpha)^T B^\alpha, U\rangle \\ &= \rho\langle U, U\rangle + \mu\langle B^\alpha U, B^\alpha U\rangle + \mu\langle U(B^\alpha)^T, U(B^\alpha)^T\rangle \\ &= \rho\|U\|_2^2 + \mu\|B^\alpha U\|_2^2 + \mu\|U(B^\alpha)^T\|_2^2 > 0,\end{aligned}$$

where we use the condition $\rho > 0, \mu > 0, U \neq 0$, and symmetricity of B^α.

The D-subproblem has two cases and these problems are same with $\mathbf{prox}_{\frac{1}{\mu}\circ|\cdot|}(\nabla^\alpha U^n + R^n)$. The difference of anisotropic and isotropic case is whether it's decoupled or not of (D_x, D_y). Therefore for anisotropic case, D-subproblem is solved by

$$\begin{aligned} D_x^{n+1} &= \max\left(K_x^n - \frac{1}{\mu}, 0\right)\operatorname{sign}(K_x^n) \\ D_y^{n+1} &= \max\left(K_y^n - \frac{1}{\mu}, 0\right)\operatorname{sign}(K_y^n)\end{aligned} \tag{3.47}$$

where

$$K^n = (K_x^n, K_y^n) = (\nabla_x^\alpha U^n + R_x^n, \nabla_y^\alpha U^n + R_y^n).$$

And for isotropic case, the solution of D-subproblem is

$$\begin{aligned} D_x^{n+1} &= \max\left(S^n - \frac{1}{\mu}, 0\right)\frac{\nabla_x^\alpha U^n + R_x^n}{S^n} \\ D_y^{n+1} &= \max\left(S^n - \frac{1}{\mu}, 0\right)\frac{\nabla_y^\alpha U^n + R_y^n}{S^n}\end{aligned} \tag{3.48}$$

where

$$S^n = \sqrt{(\nabla_x^\alpha U^n + R_x^n)^2 + (\nabla_y^\alpha U^n + R_y^n)^2}.$$

In conclusion, by applying split Bregman iteration and conjugate gradient method to (3.39), we get the following algorithm for the anisotropic and isotropic fractional TV model:

Algorithm 5 the minimization algorithm for (3.39)

Set error : $e > 0, \mu > 0$.
Inputs $U^0 = \eta,\ D_x^0 = D_y^0 = B_x^0 = B_y^0 = 0,\ \rho$.
while $\dfrac{\|U^{n+1} - U^n\|_2}{\|U^n\|_2} > e$ **do**
  Get U^{n+1} of (3.46) by the conjugate gradient method
  D^{n+1} given from (3.47) or (3.48)
  $R^{n+1} = R^n + \nabla^\alpha U^{n+1} - D^{n+1}$
  $n = n + 1$
end while.

3.3 Numerical results of fractional total variational model

In this section, we present the numerical result of our fractional total variational model for Cauchy noise. Last section, we presented two models which are anisotropic and isotropic model and we examine numerical experiments of both models. We tested twelve 256×256 gray-scale test images as shown in figure 3.1. As in chapter 2, for the original image u_0, we generated the noisy image y corrupted by Cauchy noise n which follows the Cauchy distribution $\mathcal{C}(0, \gamma)$:

$$y = u_0 + n = u_0 + \gamma \frac{\eta_1}{\eta_2},$$

where we used proposition 2.1.9 and random variables η_1, η_2 following the normal distribution $N(0, 1)$ and tested the noise levels $\gamma = 5$ and 10.

3.3.1 Parameter and termination condition

Our fractional TV for Cauchy noise model has 3 parameters: ρ, λ, α. According theorem 3.2.7, ρ should be bigger than $\dfrac{\lambda}{\gamma^2}$, so, we set $\rho = \dfrac{\lambda}{\gamma^2} + 10^{-4}$. We set the α in increments of 0.2 from 1 to 1.8 and in order to find the optimal λ for each α value, we tested the λ in increments of 5 from 10 to 80.

For termination condition, we used the energy convergence. We set energy as the target functional:

$$E(U) = \begin{cases} |\nabla_x^\alpha U| + |\nabla_y^\alpha U| + \dfrac{\lambda}{2}\langle \log(\gamma^2 + U - F)^2, \mathbf{1}\rangle & \text{for anisotropic case} \\ \sqrt{(\nabla_x^\alpha U)^2 + (\nabla_y^\alpha U)^2} + \dfrac{\lambda}{2}\langle \log(\gamma^2 + U - F)^2, \mathbf{1}\rangle & \text{for isotropic case} \end{cases} \tag{3.49}$$

And set termination condition as:

$$\frac{|E(U^{k+1}) - E(U^k)|}{|E(U^{k+1})|} < 5 \times 10^{-5}. \tag{3.50}$$

We present the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [53] for image recovery comparison which are defined as follows:

$$\text{PSNR} = 10 \log_{10}\left(\frac{255^2}{\frac{1}{mn}\|U - U_0\|_2^2}\right), \quad \text{SSIM} = \frac{(2\mu_U\mu_{U_0} + C_1)(2\sigma_{UU_0} + C_2)}{(\mu_U^2 + \mu_{U_0}^2 + C_1)(\sigma_U^2 + \sigma_{U_0}^2 + C_2)},$$

where U is the restored image, U_0 is the original image, $\mu_U, \mu_{U_0}, \sigma_U, \sigma_{U_0}$ are the means and standard deviations of U, U_0 respectively, σ_{UU_0} is the cross-covariance for U, U_0 and C_1, C_2 are positive constants.

Because of non-convexity of our fractional TV model, the result of our algorithm may depend on the initial points. Mei et. al. [34] used initial point as $\max(0, \min(255, F))$ which means that they knew the maximum value and the minimum value of images. But we assume that we don't know the maximum and minimum values of images. Instead, we applied a 3×3 median filter to a noisy image F ($median(F)$) and get a minimum value and maximum value of $median(F)$. As a result, we used following initial point:

$$V^0 = \max(m, \min(F, M))$$

where m and M are minimum and maximum values of $median(F)$, respectively.

Figure 3.1: Test images from left to right, from top to bottom: Barbara, Boat, Cameraman, Couple, House, Man, Mandrill, Peppers, Plane, Synthetic1, Synthetic2

3.3.2 Experimental results

First, we analysis anisotropic case result for noise intensity $\gamma = 5$. As we stated previous section, we solve anisotropic case in two ways (proximity operator and split Bregman iteration) and compare the results. According to the table 3.1, roughly, the higher order α, the higher PSNR and SSIM in split Bregman iteration results. Especially, Barbara, Boat, Couple and Man images have highest PSNR and SSIM in $\alpha = 1.8$. But Peppers, Synthetic1 and Synthetic2 images have highest PSNR and SSIM in $\alpha = 1$. Couple, Peppers, and Cameraman images are difficult to distinguish the difference in image quality according to a particular order. Figure 3.2 shows the results of anisotropic case using split Bregman for several α. The PSNR values of images according to alpha change are not so different, but there is a difference in how it is restored. For $\alpha = 1$, the recovered image looks very clean with little noise, but, the texture in grass is removed and three-dimensional appearance of car is lost. Overall, the image is restored like a cartoon. As α increases, the reconstructed images tends to contain texture, while noise tends not to be removed cleanly. Boat images (figure 3.3) results show similar characteristic with House. Especially, if you look at the ship's flagpole, you can see that the flagpole appears clearer as the α increases. Ocean waves also clearly visible as α increases.

From the table 3.2, we can see that the use of the proximity algorithm produce more trendy which is image quality difference due to α than the use of the split Bregman. In most images, the results of proximity algorithm have better quality than that of split Bregman itaeration. Figure 3.4 is the Barbara results of anisotropic case using proximity algorithm. You can see that the wrinkles of women's pants and tablecloth pattern in figure 3.4 become clearer as α increases. Figure 3.5 also gives the results of anisotropic case using proximity algorithm. In this figure, we can see that large α gives better detail of cloud and mountain valleys.

Now we analyze isotropic fractional TV results for noise intensity $\gamma = 5$. Table 3.3 shows the PSNR and SSIM values of isotropic fractional TV by using split Bregman method for noise intensity $\gamma = 5$. In Table 3.3, PSNR values were the highest when the α was not 1 except Peppers and synthetic2. Also, all images except cameraman, House, and Peppers showed the highest

	PSNR					SSIM				
α	1	1.2	1.4	1.6	1.8	1	1.2	1.4	1.6	1.8
Barbara	29.01	29.11	29.19	29.32	**29.62**	0.8398	0.8253	0.8245	0.8277	**0.8460**
Boat	28.99	29.67	29.67	29.85	**29.98**	0.8564	0.8458	0.8397	0.8455	**0.8650**
Cameraman	28.48	28.68	28.49	28.23	**28.63**	**0.8352**	0.8005	0.7905	0.7923	0.8142
Couple	29.28	29.39	29.40	29.48	**29.54**	0.8553	0.8439	0.8439	0.8474	**0.8615**
House	28.77	29.24	29.30	29.26	**29.38**	**0.8673**	0.8488	0.8506	0.8447	0.8669
Lena	30.63	30.69	30.57	30.83	**31.47**	**0.8819**	0.8453	0.8343	0.8422	0.8769
Man	28.25	28.85	28.92	28.93	**29.11**	0.8491	0.8522	0.8524	0.8516	**0.8662**
Mandrill	25.69	26.97	**26.98**	26.85	26.23	0.7416	0.8090	**0.8131**	0.8123	0.7890
Peppers	**30.50**	30.25	30.13	30.19	30.48	**0.8731**	0.8427	0.8347	0.8396	0.8694
Plane	29.68	30.11	30.26	30.29	**30.61**	**0.8866**	0.8358	0.8365	0.8323	0.8690
Synthetic1	**37.84**	33.91	33.90	34.21	35.92	**0.9116**	0.7907	0.7855	0.7930	0.8561
Synthetic2	**37.06**	33.84	33.60	33.26	34.34	**0.9129**	0.8122	0.8045	0.7907	0.8544

Table 3.1: PSNR and SSIM results of anisotropic fractional TV by using split Bregman iteration ($\gamma = 5$)

SSIM values when the α was not 1. In particular, most images showed the highest PSNR values when the α was 1.2. On the other hand, SSIM seems to have no particularly good α value. Figure 3.6 shows Couple image denoising results of isotropic fractional TV by using split Bregman. As the α value increases, the man's eyes in figure 3.6 become clearer and the three dimensionality of the image is revived but round artifacts tend to increase. Mandrill results in figure 3.7 also shows a similar tendency as Couple figure 3.6. Especially in the monkey's fur, you can see that the texture of the hair comes clear as the α increases.

Table 3.4 shows the results of isotropic fractional TV by using proximity algorithm for noise intensity $\gamma = 5$. Compared to the split Bregman, proximity algorithm has the maximum PSNR values at higher α values. Maximum PSNR and SSIM appeared in all images at $\alpha \neq 1$ except for Cameraman image. Also in all images except Boat, Couple, House, and Plane, maximum PSNR and SSIM appeared at the same α values.

Given the improved quality of the images of split Bregman iteration and that of proximity algorithm, the proximity algorithm gives better results. Based on this result, for the case of $\gamma = 10$, we only presented proximity algorithm results. Table 3.5 and 3.6 show the PSNR and SSIM results of

	PSNR					SSIM				
α	1	1.2	1.4	1.6	1.8	1	1.2	1.4	1.6	1.8
Barbara	28.73	28.90	29.37	29.36	**29.70**	0.8208	0.8188	0.8385	0.8352	**0.8508**
Boat	29.22	29.38	29.88	29.64	**30.09**	0.8363	0.8348	0.8568	0.8481	**0.8699**
Cameraman	28.50	28.58	28.47	28.73	**28.83**	0.7975	0.7900	0.8158	0.7994	**0.8306**
Couple	29.10	29.26	29.67	29.47	**29.71**	0.8427	0.8410	0.8584	0.8503	**0.8662**
House	28.86	29.03	29.27	29.16	**29.41**	0.8423	0.8388	0.8576	0.8522	**0.8740**
Lena	29.97	30.23	31.04	30.73	**31.60**	0.8302	0.8286	0.8599	0.8479	**0.8864**
Man	28.40	28.69	29.01	29.02	**29.12**	0.8425	0.8471	0.8606	0.8603	**0.8680**
Mandrill	26.90	**26.94**	26.74	26.72	26.31	0.8059	**0.8093**	0.8036	0.8092	0.7881
Peppers	29.87	29.89	30.54	30.42	**30.85**	0.8387	0.8308	0.8585	0.8535	**0.8807**
Plane	29.35	29.23	30.24	30.26	**30.71**	0.8189	0.7945	0.8508	0.8463	**0.8816**
Synthetic1	32.86	32.82	34.62	34.54	**35.69**	0.7750	0.7545	0.8191	0.8165	**0.8586**
Synthetic2	32.88	32.84	34.21	33.77	**35.09**	0.7967	0.7798	0.8354	0.8233	**0.8718**

Table 3.2: PSNR and SSIM results of anisotropic fractional TV by using proximity algorithm ($\gamma = 5$)

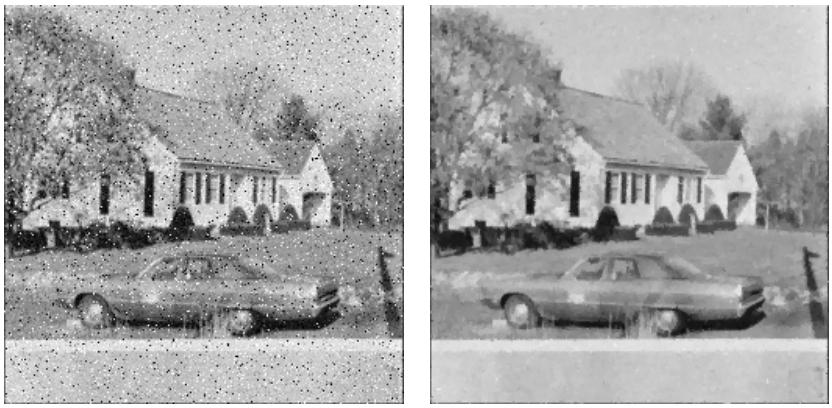
anisotropic and isotropic total variation, respectively. In the case of $\gamma = 10$, the maximum PSNR and SSIM values occurred mostly at $\alpha = 1.2$ or 1.4. Especially, there is no image which has the maximum PSNR and SSIM values at $\alpha = 1.6$. And Mandrill image is the only one that has a maximum PSNR and SSIM values at $\alpha = 1.8$ in anisotropic fractional TV model. The table 3.6 shows that the trendy of α was more clear in isotropic case. According to the table 3.6, there is no image with maximum PSNR and SSIM value when alpha is 1.6 or higher. Figure 3.8 shows Cameraman denoising result for noise intensity $\gamma = 10$. As the α value increases, the texture and volume of the recovered Cameraman images are improved. On the other hand, you can see that artifacts on recovered image due to excessive smoothing ($\alpha \geq 1.6$) on strong noise ($\gamma = 10$). Finally, figure 3.9 shows Mandrill image results for isotropic case using proximity algorithm with noise intensity $\gamma = 10$. As in figure 3.8, Mandrill image results also show better texture of the monkey's hair as α increases, but at the same time the circle artifacts grow.

	PSNR					SSIM				
α	1	1.2	1.4	1.6	1.8	1	1.2	1.4	1.6	1.8
Barbara	29.24	29.40	29.30	**29.44**	29.26	0.8509	0.8473	0.8507	0.8503	**0.8512**
Boat	29.46	**29.90**	29.20	29.54	28.81	0.8637	0.8650	0.8625	**0.8654**	0.8581
Cameraman	27.67	**27.93**	26.73	27.06	26.23	**0.8452**	0.8282	0.8324	0.8268	0.8266
Couple	29.15	**29.60**	28.90	29.15	28.38	0.8592	**0.8638**	0.8542	0.8576	0.8437
House	28.94	**29.34**	28.65	28.74	27.82	**0.8722**	0.8682	0.8673	0.8675	0.8575
Lena	31.09	31.36	**31.53**	31.51	31.38	0.8871	0.8757	0.8954	0.8917	**0.8995**
Man	28.60	**29.06**	28.61	28.72	27.93	0.8568	**0.8633**	0.8600	0.8632	0.8522
Mandrill	25.75	**26.36**	25.50	25.86	25.24	0.7500	**0.7862**	0.7430	0.7684	0.7336
Peppers	**30.87**	30.70	30.30	30.25	29.10	**0.8870**	0.8734	0.8879	0.8805	0.8851
Plane	30.01	**30.56**	30.20	30.09	29.16	0.8891	0.8718	**0.8909**	0.8846	0.8909
Synthetic1	37.87	36.04	37.82	37.06	**38.16**	0.9115	0.8630	0.9119	0.8920	**0.9218**
Synthetic2	**36.84**	35.21	35.48	34.05	31.88	0.9107	0.8724	0.9156	0.8966	**0.9194**

Table 3.3: PSNR and SSIM results of isotropic fractional TV by using split Bregman iteration ($\gamma = 5$)

	PSNR					SSIM				
α	1	1.2	1.4	1.6	1.8	1	1.2	1.4	1.6	1.8
Barbara	29.27	29.20	29.41	29.49	**29.51**	0.8495	0.8506	0.8516	0.8482	**0.8539**
Boat	29.72	29.14	29.76	**29.94**	29.76	0.8659	0.8600	0.8685	0.8674	**0.8702**
Cameraman	**28.39**	27.75	27.65	27.92	28.31	**0.8441**	0.8405	0.8355	0.8278	0.8406
Couple	29.39	28.74	29.45	**29.48**	29.23	0.8630	0.8501	**0.8641**	0.8609	0.8582
House	29.09	28.61	29.19	**29.32**	29.02	0.8710	0.8676	**0.8740**	0.8711	0.8739
Lena	31.05	31.28	31.67	31.64	**31.84**	0.8818	0.8971	0.8910	0.8821	**0.8984**
Man	28.75	28.40	28.88	**29.09**	28.77	0.8602	0.8540	0.8641	**0.8669**	0.8641
Mandrill	26.20	25.40	26.04	**26.33**	25.44	0.7763	0.7282	0.7722	**0.7924**	0.7457
Peppers	30.82	**30.83**	30.70	30.43	30.56	0.8805	**0.8943**	0.8838	0.8745	0.8884
Plane	30.25	30.21	30.61	30.67	**30.69**	0.8728	**0.8944**	0.8795	0.8748	0.8940
Synthetic1	36.90	**38.71**	36.99	36.35	37.35	0.8903	**0.9325**	0.8923	0.8734	0.8999
Synthetic2	36.23	**37.07**	35.55	34.83	35.08	0.8956	**0.9341**	0.8983	0.8814	0.9079

Table 3.4: PSNR and SSIM results of isotropic fractional TV by using proximity algorithm ($\gamma = 5$)

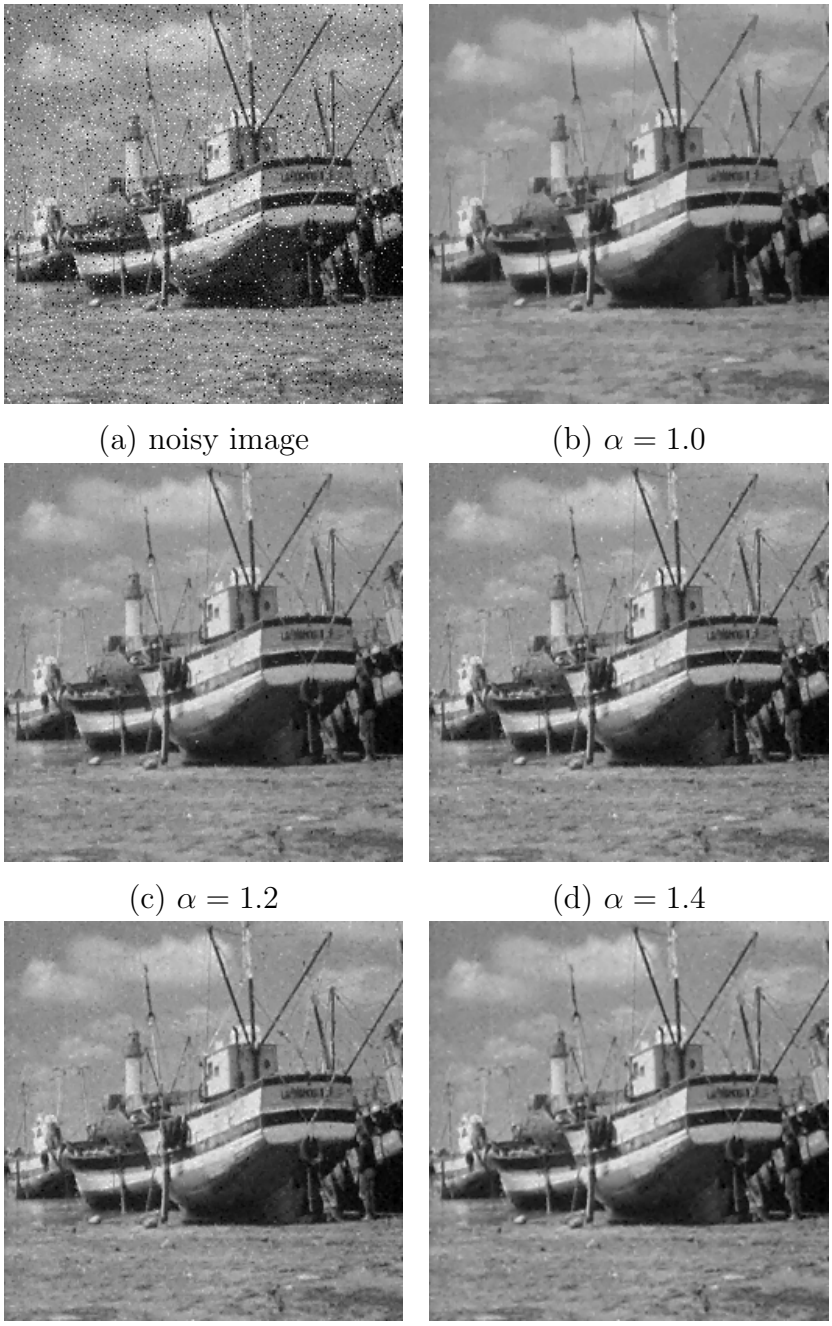


(a) noisy image (b) $\alpha = 1.0$

(c) $\alpha = 1.2$ (d) $\alpha = 1.4$

(e) $\alpha = 1.6$ (f) $\alpha = 1.8$

Figure 3.2: House image results of anistropic case using split Bregman ($\gamma = 5$)

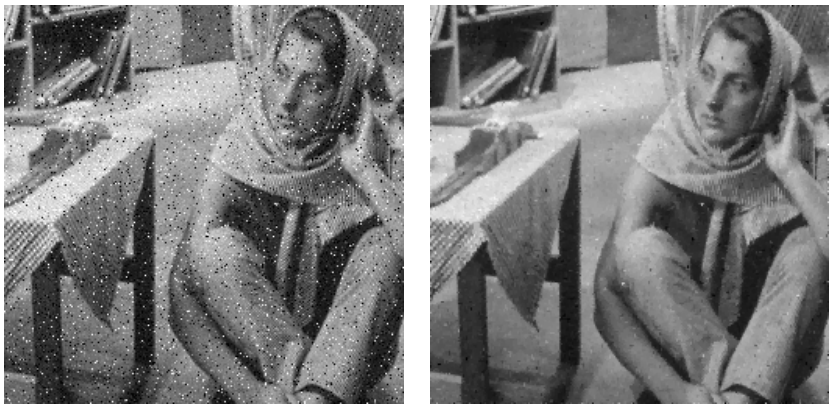


(a) noisy image (b) $\alpha = 1.0$

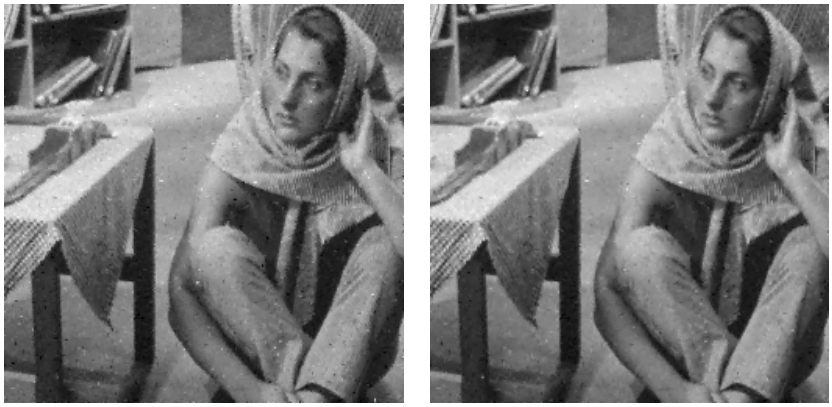
(c) $\alpha = 1.2$ (d) $\alpha = 1.4$

(e) $\alpha = 1.6$ (f) $\alpha = 1.8$

Figure 3.3: Boat image results of anistropic case using split Bregman ($\gamma = 5$)



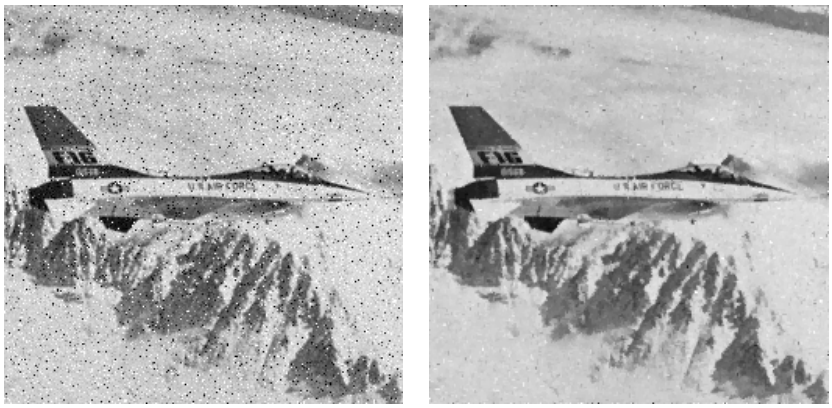
(a) noisy image (b) $\alpha = 1.0$



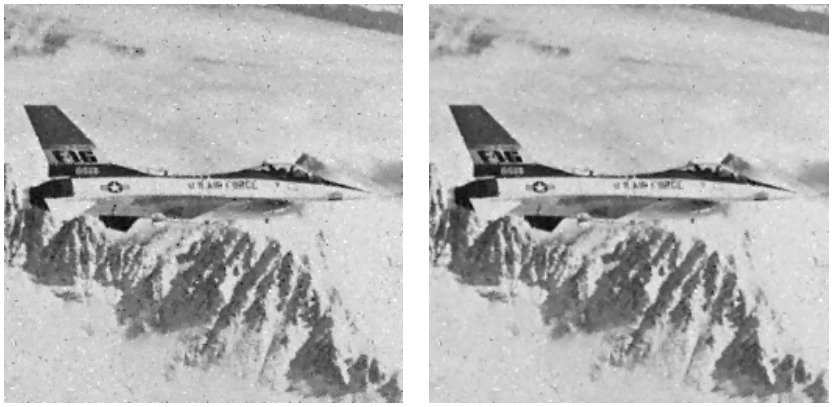
(c) $\alpha = 1.2$ (d) $\alpha = 1.4$

(e) $\alpha = 1.6$ (f) $\alpha = 1.8$

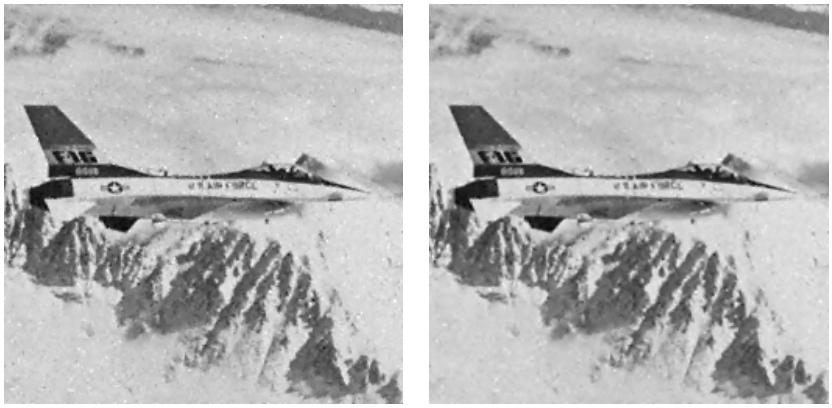
Figure 3.4: Barbara image results of anisotropic case using proximity algorithm ($\gamma = 5$)



(a) noisy image (b) $\alpha = 1.0$

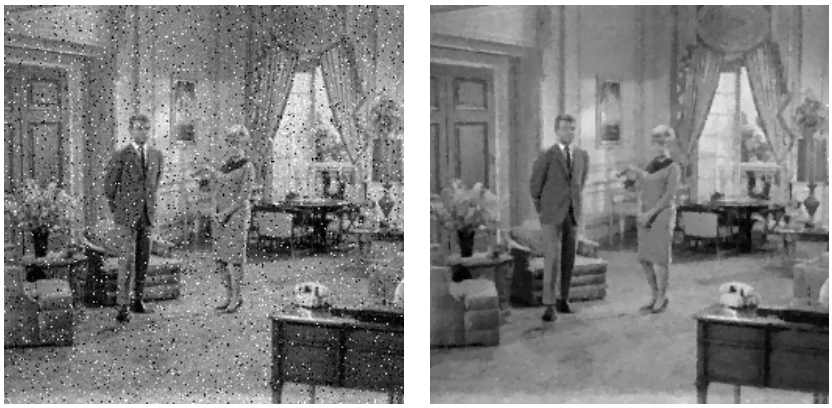


(c) $\alpha = 1.2$ (d) $\alpha = 1.4$



(e) $\alpha = 1.6$ (f) $\alpha = 1.8$

Figure 3.5: Plane image results of anisotropic case using proximity algorithm ($\gamma = 5$)

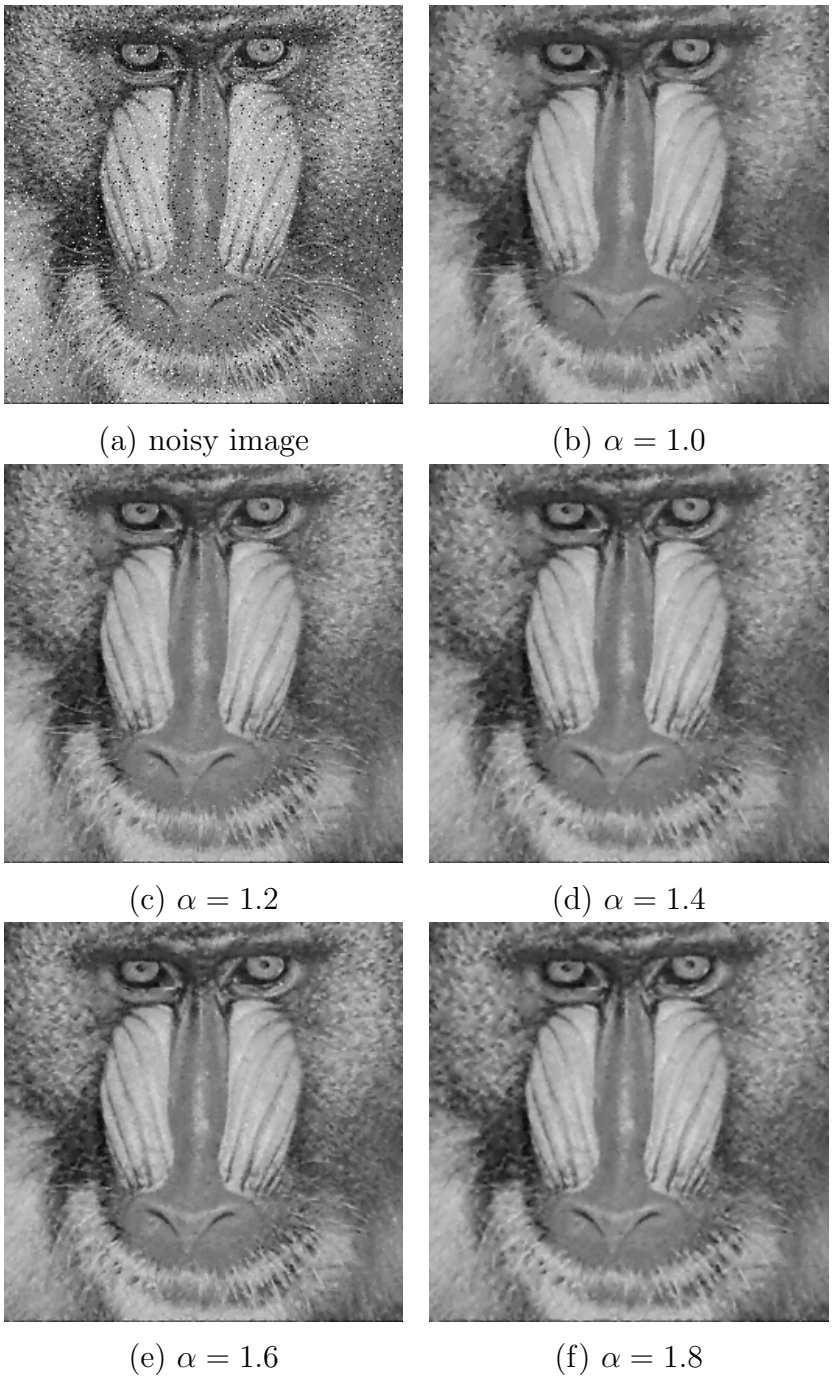


(a) noisy image (b) $\alpha = 1.0$

(c) $\alpha = 1.2$ (d) $\alpha = 1.4$

(e) $\alpha = 1.6$ (f) $\alpha = 1.8$

Figure 3.6: Couple image results of isotropic case using split Bregman iteration ($\gamma = 5$)



(a) noisy image (b) $\alpha = 1.0$

(c) $\alpha = 1.2$ (d) $\alpha = 1.4$

(e) $\alpha = 1.6$ (f) $\alpha = 1.8$

Figure 3.7: Mandrill image results of isotropic case using split Bregman iteration ($\gamma = 5$)

	PSNR					SSIM				
α	1	1.2	1.4	1.6	1.8	1	1.2	1.4	1.6	1.8
Barbara	27.67	27.89	**27.94**	27.82	27.62	0.7987	**0.8043**	0.7991	0.7852	0.7682
Boat	27.38	27.21	**27.61**	27.34	27.15	0.7933	0.7891	**0.7966**	0.7731	0.7542
Cameraman	26.18	25.69	**26.60**	26.45	26.24	**0.7928**	0.7889	0.7857	0.7355	0.6914
Couple	27.20	27.09	**27.48**	27.26	27.07	0.7822	0.7778	**0.7887**	0.7715	0.7564
House	26.88	26.82	**27.00**	26.69	26.65	**0.8084**	0.8050	0.8053	0.7760	0.7523
Lena	28.90	29.14	**29.28**	28.58	28.26	0.8373	**0.8467**	0.8422	0.8099	0.7774
Man	26.60	26.57	**26.81**	26.66	26.59	0.7821	0.7794	**0.7861**	0.7723	0.7633
Mandrill	24.73	24.48	24.66	24.72	**24.81**	0.6600	0.6395	0.6588	0.6620	**0.6750**
Peppers	28.47	**28.64**	28.58	28.24	27.95	0.8436	**0.8518**	0.8442	0.8173	0.7835
Plane	27.88	27.87	**28.32**	28.05	27.77	0.8415	**0.8442**	0.8281	0.7978	0.7586
Synthetic1	35.63	**36.57**	34.10	33.56	30.26	0.8907	**0.9177**	0.8469	0.7880	0.6462
Synthetic2	34.56	**34.61**	33.29	32.07	30.18	0.8899	**0.9159**	0.8720	0.8026	0.6883

Table 3.5: PSNR and SSIM results of anisotropic fractional TV by using proximity algorithm ($\gamma = 10$)

	PSNR					SSIM				
α	1	1.2	1.4	1.6	1.8	1	1.2	1.4	1.6	1.8
Barbara	27.76	27.84	**27.99**	27.81	27.64	0.8039	**0.8060**	0.8055	0.7954	0.7814
Boat	27.06	26.60	**27.28**	26.45	27.08	0.7874	0.7759	**0.7906**	0.7651	0.7676
Cameraman	25.77	25.97	**26.30**	26.22	24.99	0.7943	**0.7971**	0.7873	0.7598	0.7168
Couple	**26.93**	26.63	26.49	26.56	26.65	**0.7741**	0.7622	0.7622	0.7602	0.7553
House	**26.73**	26.38	26.71	26.57	25.98	**0.8069**	0.7949	0.8001	0.7861	0.7607
Lena	29.09	29.32	**29.38**	28.82	29.18	0.8470	**0.8535**	0.8479	0.8257	0.8075
Man	26.55	26.34	**26.82**	26.67	26.64	0.7800	0.7715	**0.7870**	0.7825	0.7781
Mandrill	24.40	24.13	**24.64**	24.19	24.19	0.6359	0.6109	**0.6574**	0.6388	0.6459
Peppers	**28.86**	28.35	27.94	28.07	27.55	0.8531	**0.8551**	0.8469	0.8320	0.8087
Plane	27.71	27.35	**27.84**	26.90	26.93	**0.8484**	0.8442	0.8403	0.8088	0.7839
Synthetic1	36.46	**37.32**	36.03	34.45	33.15	0.9142	**0.9361**	0.8970	0.8313	0.7652
Synthetic2	**35.19**	34.65	32.39	31.72	29.74	0.9098	**0.9318**	0.8953	0.8340	0.7709

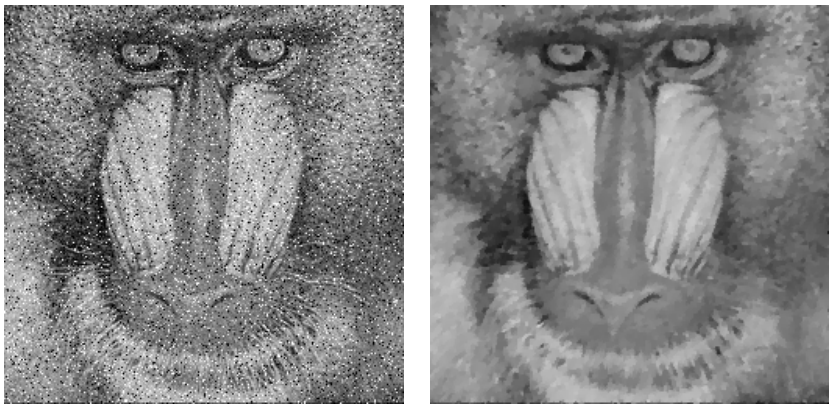
Table 3.6: PSNR and SSIM results of isotropic fractional TV by using proximity algorithm ($\gamma = 10$)

(a) noisy image (b) $\alpha = 1.0$

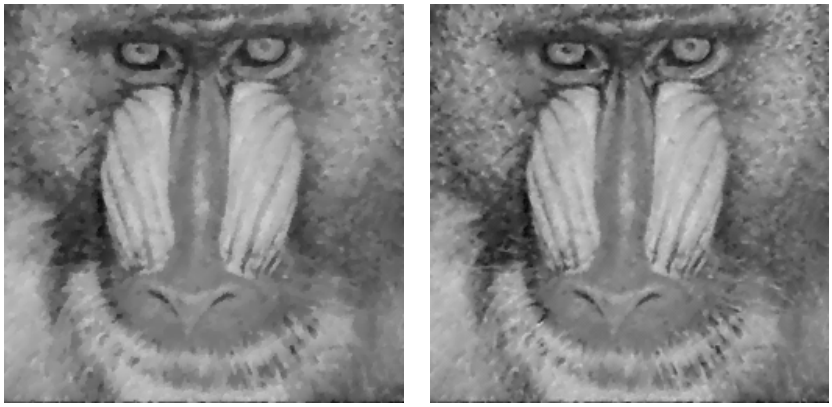
(c) $\alpha = 1.2$ (d) $\alpha = 1.4$

(e) $\alpha = 1.6$ (f) $\alpha = 1.8$

Figure 3.8: Cameraman image results of anisotropic case using proximity algorithm ($\gamma = 10$)



(a) noisy image (b) $\alpha = 1.0$



(c) $\alpha = 1.2$ (d) $\alpha = 1.4$

(e) $\alpha = 1.6$ (f) $\alpha = 1.8$

Figure 3.9: Mandrill image results of isotropic case using proximity algorithm ($\gamma = 10$)

Chapter 4

Nuclear norm minimization and Cauchy noise denoising model

In this chapter, we examine nuclear norm and weighted nuclear norm which is more adaptive norm for the rank minimization and explain why weighted model is appropriate. Then, we apply weighted nuclear norm to Cauchy noise removal and solve our model based on weighted nuclear norm by nonconvex ADMM. Lastly, we present some result of our model which is better recovery quality than the state-of-the-art models in Cauchy noise removal.

4.1 Weighted Nuclear Norm

In general, non-local methods have much better performances than local methods in the field of image noise removal. It is famous for its non-local methods such as non-local means(NL-means)[5], Block matching and 3D filtering(BM3D)[13], non-local total variation(NLTV)[18] and Weighted Nuclear Norm Minimization(WNNM)[22], etc. Among them, WNNM performs very well. Therefore, we start to examine WNNM and apply to Cauchy noise removal. Before we propose our denoising model, in this section, we first look at weighted nuclear norm.

4.1.1 Weighted Nuclear Norm and Its Applications

First, we give the definition.

Definition 4.1.1. *The nuclear norm of a matrix $X \in \mathbb{R}^{n\times m}$ is the sum of its singular values.*

$$\|X\|_* = \sum_i \sigma_i(X) \tag{4.1}$$

where $\sigma_i(X)$ is the i-th singular value of the matrix X.
And the weighted nuclear norm of a matrix X is defined as follows:

$$\|X\|_{w,*} = \sum_i w_i\sigma_i(X), \tag{4.2}$$

where $w = [w_1, w_2, \ldots, w_k], w_i > 0$ for $1 \le i \le k$ and $k = \min(n, m)$.

The nuclear norm is a norm but, in general, weighted nuclear norm is not a norm.

Proposition 4.1.2. *The nuclear norm $\|\cdot\|_*$ is a norm.*

Proof. Two conditions are easily verified.
First, if $\|X\|_* = 0$. Then $\sigma_i(X) = 0$ for all i, so, $X = 0$.
Second, $\|cX\|_* = \sum_i \sigma_i(cX) = \sum_i |c|\sigma_i(X) = |c|\|X\|_*$
Last, most non-trivial part is triangular inequality. i.e.

$$\|X + Y\|_* \le \|X\|_* + \|Y\|_* \tag{4.3}$$

To show triangular inequality, we use following equation:

$$\sup_{\sigma_1(Q)\le 1} \langle Q, X\rangle = \sup_{\sigma_1(Q)\le 1} \mathrm{Tr}(Q^T X) = \sum_i \sigma_i(X) = \|X\|_*. \tag{4.4}$$

If (4.4) is satisfied, then

$$\begin{aligned}\|X + Y\|_* &= \sup_{\sigma_{\max}(R)\le 1} \langle R, X + Y\rangle \\ &\le \sup_{\sigma_{\max}(R)\le 1} \langle R, X\rangle + \sup_{\sigma_{\max}(R)\le 1} \langle R, Y\rangle = \|X\|_* + \|Y\|_*\end{aligned}$$

This proves the triangular inequality.
Now, we have to show (4.4). Let $X = U\Sigma V^T = \sum_i \sigma_i u_i v_i^T$ be the singular value decomposition(SVD) of X, and define $Q = UV^T = UIV^T$. Then by construction, $\sigma_1(Q) = 1$ and

$$\begin{aligned}\langle Q, X\rangle &= \langle UV^T, U\Sigma V^T\rangle \\ &= \mathrm{Tr}(VU^TU\Sigma V^T) \\ &= \mathrm{Tr}(V^TVU^TU\Sigma) = \mathrm{Tr}(\Sigma) = \sum_i \sigma_i(X).\end{aligned}$$

Therefore,

$$\sum_i \sigma_i(X) \leq \sup_{\sigma_1(Q)\leq 1} \langle Q, X\rangle.$$

Now let us prove the other direction:

$$\begin{aligned}\sup_{\sigma_1(Q)\leq 1} \langle Q, X\rangle &= \sup_{\sigma_1(Q)\leq 1} \langle Q, U\Sigma V^T\rangle = \sup_{\sigma_1(Q)\leq 1} \mathrm{Tr}(Q^TU\Sigma V^T) \\ &= \sup_{\sigma_1(Q)\leq 1} \mathrm{Tr}(V^TQ^TU\Sigma) = \sup_{\sigma_1(Q)\leq 1} \langle U^TQV, \Sigma\rangle \\ &= \sup_{\sigma_1(Q)\leq 1} \sum_i (U^TQV)_{ii}\sigma_i \leq \sup_{\sigma_1(Q)\leq 1} \sum_i \sigma_{\max}(Q)\sigma_i = \sum_i \sigma_i\end{aligned}$$

which proves (4.4). □

In the above proof, we used the SVD ($X = U\Sigma V^T$) and Σ is diagonal matrix consist of singular values of X. In this thesis, we naturally assume singular values sorted by descending order in Σ.
By using that every norm is convex operator, we can get the following corollary.

Corollary 4.1.3. *The nuclear norm $\|\cdot\|_*$ is convex.*

Because of convexity, nuclear norm is used for low-rank matrix approximation. Low rank matrix approximation aims to find a matrix with a low-rank that approximates to the given matrix. Low-rank matrix approximation is used in many places, such as data compression [24], linear models fitting [33], subspace segmentation [31], collaborative filtering [47]. The nuclear

norm minimization minimize L_1-norm of singular values which is slightly different from low-rank. But according to Cai et al. [6], for $Y \in \mathbb{R}^{n\times m}$, the solution of the minimization problem

$$\hat{X} = \arg\min_X \frac{1}{2}\|Y - X\|_2^2 + \lambda\|X\|_*$$

is given by

$$\hat{X} = U\text{soft}_\lambda(\Sigma)V^T,$$

where $Y = U\Sigma V^T$ is the singular value decomposition(SVD) of Y and soft_λ is soft thresholding operator by λ. However, there are disadvantages of nuclear norm minimization problem. For example, not all singular values have the same importance. The larger singular values tend to have more principal data. Therefore, rather than soft thresholding all singular values by the same value as nuclear norm minimization, it is more reasonable to reduce the bigger singular value by the smaller value. By applying this tendency, Gu et al.[22] proposed weighted nuclear norm minimization model as follows:

$$\hat{X} = \arg\min_{X\in\mathbb{R}^{n\times m}} \|X\|_{w,*} + \frac{1}{2}\|Y - X\|_2^2 \tag{4.5}$$

where $Y \in \mathbb{R}^{n\times m}$ and $w = [w_1, \ldots, w_k], k = \min(n, m), w_i > 0$. Due to the presence of weights, WNNM is not convex in general, but under some conditions it is possible to obtain useful theorems [22].

Theorem 4.1.4. *Let us denote $Y = U\Sigma V^T$ as the SVD of $Y \in \mathbb{R}^{n\times m}$. The solution of (4.5), $\hat{X}$ is given by $\hat{X} = U\hat{B}V^T$, where $\hat{B}$ is the solution of the following minimization problem:*

$$\hat{B} = \arg\min_B \|B\|_{w,*} + \frac{1}{2}\|\Sigma - B\|_2^2. \tag{4.6}$$

To prove the theorem 4.1.4, we need the following lemma.

Lemma 4.1.5. *If $M_1^T M_2 = 0$ for $M_1, M_2 \in \mathbb{R}^{n\times m}$, then the following equa-*

tions holds:

$$1) \quad \|M_1 + M_2\|_{w,*} \geq \|M_1\|_{w,*} \tag{4.7}$$
$$2) \quad \|M_1 + M_2\|_F \geq \|M_1\|_F. \tag{4.8}$$

For those who interest in the proof of the lemma 4.1.5, see [22].

Now we are ready to prove theorem 4.1.4.

(Proof of Theorem 4.1.4). For any $X \in \mathbb{R}^{n\times m}$, we can write $X = UA + U_\perp \tilde{A}$ where $U_\perp$ is the set of orthogonal bases of the complementary space of U and A and $\tilde{A}$ are components of X in subspace U and $U_\perp$, respectively. Then $U^T U_\perp = 0$ and

$$\begin{aligned}
&\|Y - X\|_F^2 + \|X\|_{\mathbf{w},*} \\
&= \|U\Sigma V^T - UA - U_\perp \tilde{A}\|_F^2 + \|UA + U_\perp \tilde{A}\|_{\mathbf{w},*} \\
&\geq \|U\Sigma V^T - UA\|_F^2 + \|UA\|_{\mathbf{w},*}
\end{aligned}$$

In here, we used lemma 4.1.5. Then split $A^T = VB^T + V\tilde{B}^T$ and use lemma 4.1.5 again, we get

$$\|Y - X\|_F^2 + \|X\|_{\mathbf{w},*} \geq \|U\Sigma V^T - UBV^T\|_F^2 + \|UBV^T\|_{\mathbf{w},*}$$

after we use the property that orthogonal matrices are not change frobenius norm and weighted nuclear norm, then

$$\begin{aligned}
\|Y - X\|_F^2 + \|X\|_{\mathbf{w},*} &\geq \|\Sigma - B\|_F^2 + \|B\|_{\mathbf{w},*} \\
&\geq \|\Sigma - \hat{B}\|_F^2 + \|\hat{B}\|_{\mathbf{w},*}
\end{aligned}$$

Therefore, the solution of (4.5) can be obtained from $\hat{X} = U\hat{B}V^T$. □

For the descending order weights (i.e., $w_1 \geq \cdots \geq w_k \geq 0$), we can get a globally optimal solution by soft thresholding.

Theorem 4.1.6. *Assume weights* $\mathbf{w} = [w_1, \cdots, w_k]$ *satisfy* $w_1 \geq \cdots \geq w_k \geq 0$ *and , then we can get a globally optimal solution of* (4.5) *as follows:*

$$\hat{X} = U soft_{\mathbf{w}}(\Sigma) V^T \tag{4.9}$$

where $Y = U\Sigma V^T$ *is SVD of* Y *and* $soft_{\mathbf{w}}(\Sigma)_{ii} = \max(\Sigma_{ii} - w_i, 0)$.

By using following lemma, we prove theorem 4.1.6.

Lemma 4.1.7. *Assume* $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$, $M_{11} \in \mathbb{R}^{m\times m}$, $M_{22} \in \mathbb{R}^{n\times n}$, *and weight* $\mathbf{w} = [w_1, \ldots, w_{m+n}]$ *are non-negative and in a descending order (i.e.* $w_1 \geq \cdots \geq w_{m+n} \geq 0$*), then we have*

$$\|M\|_{\mathbf{w},*} \geq \|M_{11}\|_{\mathbf{w_1},*} + \|M_{22}\|_{\mathbf{w_2},*} \tag{4.10}$$

where $\mathbf{w_1} = [w_1, \ldots, w_m]$ *and* $\mathbf{w_2} = [w_{m+1}, \ldots, w_{m+n}]$.

The proof of the lemma 4.1.7 can be found in [31].

(Proof of Theorem 4.1.6). Enough to show that the solution of (4.6) is $\hat{B} = S_{\mathbf{w}}(\Sigma)$. Let D_B is a diagonal matrix with $(D_B)_{ii} = B_{ii}$. Then according to the definition of Frobenius norm,

$$\|\Sigma - B\|_F^2 = \|\Sigma - D_B - (B - D_B)\|_F^2 \geq \|\Sigma - D_B\|_F^2.$$

And by using lemma 4.1.7, iteratively, we can get

$$\|B\|_{\mathbf{w},*} = \|D_B + (B - D_B)\|_{\mathbf{w},*} \geq \|D_B\|_F^2.$$

Combining the two inequality, we get

$$\|\Sigma - B\|_F^2 + \|B\|_{\mathbf{w},*} \geq \|\Sigma - D_B\|_F^2 + \|D_B\|_{\mathbf{w}.*}. \tag{4.11}$$

Both Σ and D_B are diagonal matrices, we can get the solution elementwise by soft-thresholding: $\text{soft}_{w_i}(\Sigma_{ii})$. □

According to the [21], we can get the useful theorem. To do this, we observe the following von Neumanns trace inequality [37].

Lemma 4.1.8. *Let* $A, B \in \mathbb{R}^{n\times m}$. *Then,*

$$Tr(A^T B) \leq \sum_i \sigma_i(A)\sigma_i(B)$$

where $\sigma_i(M)$ is i-th singular value of M. (As we stated above, singular values of any matrix are sorted by descending order.) The equality satisfied when if there exist U and V such that

$$A = U\Sigma_A V^T \qquad B = U\Sigma_B V^T.$$

Based on the lemma 4.1.8, we get the following important theorem.

Theorem 4.1.9. *Let us denote $Y = U\Sigma V^T$ as the SVD of $Y \in \mathbb{R}^{n\times m}$ and weight $\mathbf{w} = [w_1, \ldots, w_k]$ satisfy $0 \le w_1 \le \cdots \le w_k, k = \min(n, m)$, then the global optimum of (4.5) is $\hat{X} = U soft_{\mathbf{w}}(\Sigma)V^T$.*

Proof. For $X \in \mathbb{R}^{n\times m}$, let $X = QDR^T$ be the SVD of X and denote diagonal element of D_{ii} as d_i. By the property of Frobenius norm and trace, the following derivations hold:

$$\begin{aligned}\|X\|_{\mathbf{w},*} + \frac{1}{2}\|X - Y\|_F^2 &= \frac{1}{2}Tr(Y^TY) - \mathrm{Tr}(Y^TX) + \frac{1}{2}\mathrm{Tr}(X^TX) + \sum_i w_i d_i \\ &= \frac{1}{2}\sum_i \sigma_i^2 - \mathrm{Tr}(Y^TX) + \frac{1}{2}\sum_i d_i^2 + \sum_i w_i d_i\end{aligned}$$

Based on the lemma 4.1.8, if $Q = U$ and $R = V$, then $\mathrm{Tr}(X^TY) = \sum_i \sigma_i d_i$, so,

$$\begin{aligned}\min_X \|X\|_{\mathbf{w},*} + \frac{1}{2}\|X - Y\|_F^2 &= \min_D \frac{1}{2}\sum_i \sigma_i^2 - \sum_i \sigma_i d_i + \frac{1}{2}\sum_i d_i^2 + \sum_i w_i d_i \\ &= \min_D \frac{1}{2}\sum_i (d_i - \sigma_i)^2 + w_i d_i\end{aligned}$$

Now, we can solve the above minimization by soft-thresholding:

$$d_i = \max\left(\sigma_i - w_i, 0\right), \qquad 1 \le i \le k \tag{4.12}$$

Therefore, $\hat{X} = U\mathrm{soft}_{\mathbf{w}}V^T$ is the globally optimal solution of (4.5). □

Since the larger singular values of a matrix mostly correspond to the more important subspace. In the image denoising, Gu et al. [22, 21] has proposed

a model that shrink small values for large singular values and shrink large values for small singular values. For Gaussian noise image $y \in \mathbb{R}^{N\times M}$, they stack similar patches $Y \in \mathbb{R}^{n\times m}$ of a patch, then minimize following equation

$$\underset{X\in\mathbb{R}^{n\times m}}{\arg\min} \|X\|_{w,*} + \frac{1}{2}\|Y - X\|_2^2 \tag{4.13}$$

where weights are in ascending order $0 \le w_1 \le \cdots \le w_k,\ k = \min(n, m)$. And the theorem 4.1.9 guarantee that our WNNM problem can be solved simply by soft-thresholding.

4.1.2 Iteratively Reweighted l_1 Minimization

Concretely, Gu et al. [22, 21] has proposed a weight of (4.13) as the inverse of the singular value of the true signal data (clean image data) X^*. i.e.

$$w_i = \begin{cases} \dfrac{1}{\sigma_i(X^*)} & \text{if } \sigma_i(X^*) \neq 0 \\ \infty & \text{if } \sigma_i(X^*) = 0 \end{cases} \tag{4.14}$$

But it is impossible to construct the weight (4.14) without the true signal. To approximate the weight (4.14), they used iteratively reweighting l_1 minimization algorithm [7].

In scientific or engineering problems, we often see more unknown variables than equations. When the equations are linear, we can write the problem as $y = \Phi x$ for an $x \in \mathbb{R}^n, y \in \mathbb{R}^m$ and $m < n$. Obviously, it is impossible to identify correct solution without some additional information. But in many situation, the unknown data x depends only on a small number of variables. Therefore, to find solution of the linear equation $y = \Phi x$, it is usually assumed that the sparsity of the solution x^*. Mathematical expression is as follows:

$$\min_{x\in\mathbb{R}^n} \|x\|_{l_0} \quad \text{s.t.} \quad y = \Phi x,\ y \in \mathbb{R}^m,\ m < n. \tag{4.15}$$

where $\|x\|_{l_0} = |\{i : x_i \neq 0\}|$. But l_0-norm problem is NP-hard problem [38, 14], so, it is almost impossible to solve. To solve the problem (4.15),

Candés et al. considered the weighted l_1 minimization problem:

$$\min_{x\in\mathbb{R}^n} \sum_i w_i|x_i| \quad \text{s.t.} \quad y = \Phi x, \tag{4.16}$$

where $w_i > 0$. And if the true data x^* is k-sparse (i.e. $\|x^*\|_{l_0} \leq k$) for $k \leq m$ and the weights were inverse of the true data, i.e.,

$$w_i = \begin{cases} \dfrac{1}{x_i^*} & \text{if } x_i^* \neq 0 \\ \infty & \text{if } x_i^* = 0, \end{cases} \tag{4.17}$$

then (4.16) can find the correct solution. But we cannot set weights as the inverse of x^* without knowing the true data x^*. To obtain a valid set of weights, Candés et al. proposed the following iterative algorithm:

Algorithm 6 Iterative reweighted l_1 minimization algorithm

Set $l = 1$, $w_i^{(1)} = 1, i = 1, \ldots, n$, and l_{max}.
for $l < l_{max}$ or $x^{(l)}$ converges **do**
Solve the weighted l_1 minimization problem

$$x^{(l)} = \arg\min_{x\in\mathbb{R}^n} \sum_i w_i^{(l)}|x_i| \quad \text{s.t.} \quad y = \Phi x. \tag{4.18}$$

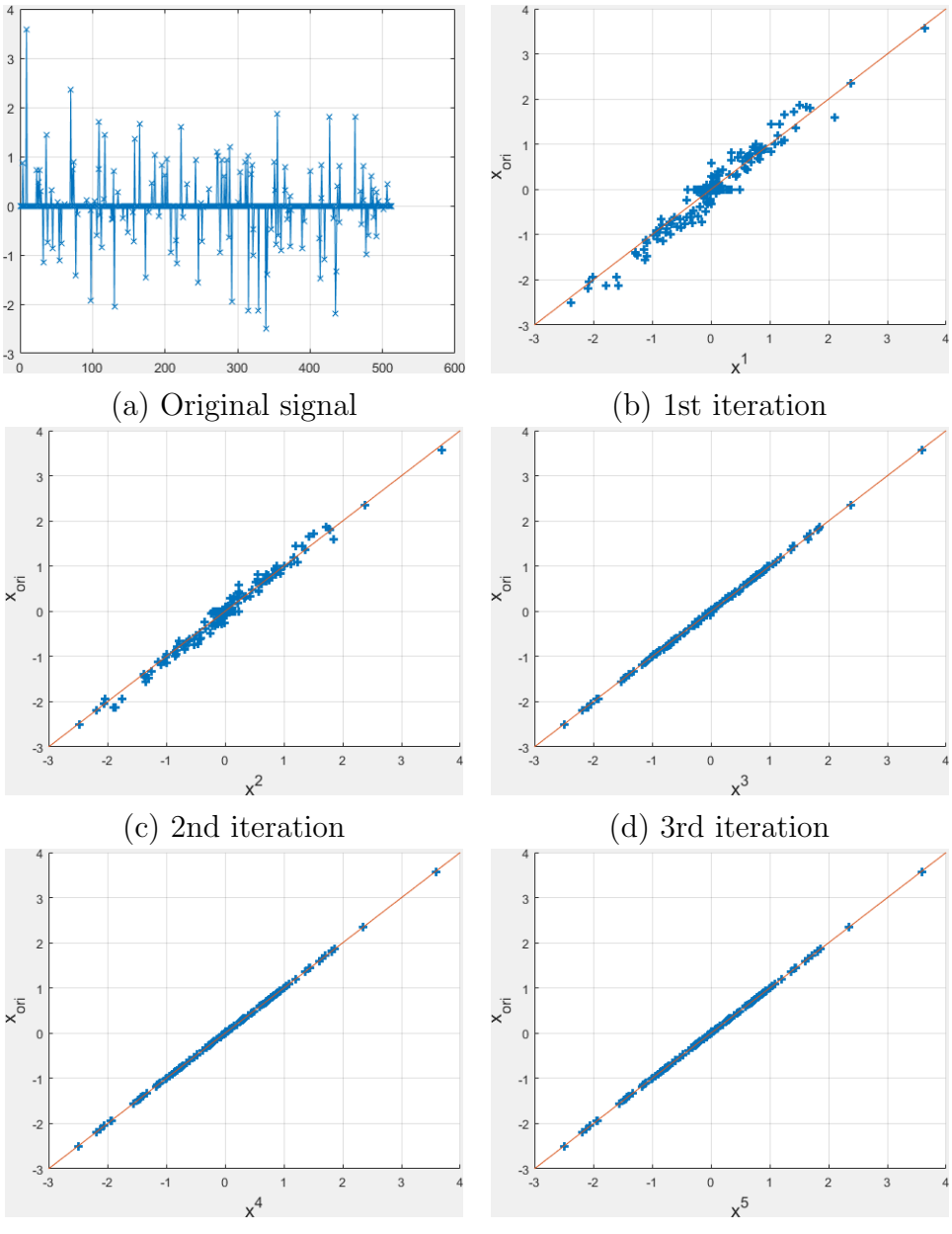
Update the weights:

$$w_i^{(l+1)} = \frac{1}{|x_i^{(l)}| + \epsilon}.$$

Update $l = l + 1$
end for

where $\epsilon > 0$ prevent numerator from dividing by zero. The above algorithm gradually finds zero coefficient locations of true signal by repeating for loop.

Figure 4.1 shows the process of sparse signal recovery by the algorithm 6. The (a) in figure 4.1 shows the original signal x_{ori}. The original data has length $n = 512$ and 130 nonzero values and 382 zero values.



(a) Original signal (b) 1st iteration

(c) 2nd iteration (d) 3rd iteration

(e) 4th iteration (f) 5th iteration

Figure 4.1: Sparse signal recovery by iterative reweighted l_1 minimization algorithm

We set the Φ as 256×512 matrix by independent standard normal entries and $\epsilon = 0.5$. Figure 4.1 (b)-(f) show scatter plots of coefficient of l-th iteration solution x^l versus the original signal coefficient. In usual l_1 norm minimization (Fig. 4.1(b)), we see that the large coefficient of x_{ori} are somewhat predicted well in x^1 but the solution x^1 and the original data are quite different. Actually, the maximum difference between x^1 and x_{ori} is $\|x^1 - x_{ori}\|_{l_\infty} = 0.5943$. As the reweighted iteration progresses, the difference between solution x_l and x_{ori} decrease and the values are $\|x^2 - x_{ori}\|_{l_\infty} = 0.3601$, $\|x^3 - x_{ori}\|_{l_\infty} = 0.0816$, $\|x^4 - x_{ori}\|_{l_\infty} = 0.0325$, and $\|x^5 - x_{ori}\|_{l_\infty} = 0.0324$. Around fourth iteration, the estimated signal almost perfectly recover the original signal.

To find optimal weights, therefore, Gu et al. applied algorithm 6 to the WNNM. For each stacked similar patches $Y \in \mathbb{R}^{n\times m}$, they iterated (4.13) by updating weights to the inverse of the singular values of solution of (4.13).

4.2 Proposed Model: Weighted Nuclear Norm For Cauchy Noise Denoising

4.2.1 Model and algorithm discription

By the powerfulness of WNNM, we introduce our variational model for Cauchy noise denoising, which adopts the weighted nuclear norm as a regularizer.

Let $y \in \mathbb{R}^{M\times N}$ be the Cauchy noise image and y_i be a patch of $\sqrt{n} \times \sqrt{n}$ in the noise image y, we grouped similar patches for each y_i according to a specific metric, then vectorized and stacked them to form a two-dimensional matrix Y_i. Figure 4.2 show the process of constructing matrix Y_i. Then, for each Y_i, we solve the following problem:

$$\arg\min_{X_i} \|X_i\|_{\mathbf{w},*} + \sum_{x\in X_i, y\in Y_i} \log\left(\gamma^2 + (x-y)^2\right) \tag{4.19}$$

where $\|X\|_{\mathbf{w},*}$ is the weighted nuclear norm of X, and we adopt the discrete version of fidelity term in (2.11).

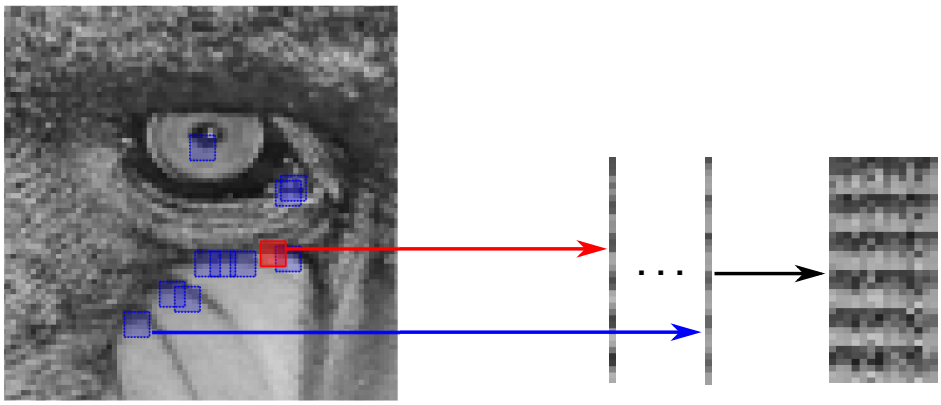


Figure 4.2: The process of constructing matrix composed of similar patches

We solve the model (4.19) by the nonconvex ADMM algorithm that stated in section 3.2.2. To apply nonconvex ADMM, we reformulate (4.19) as follows:

$$\arg\min_{X} \|X\|_{\mathbf{w},*} + \frac{\lambda}{2}\langle \log(\gamma^2 + (X-Y)^2), \mathbf{1}\rangle \tag{4.20}$$

where stacked similar image patches $Y \in \mathbb{R}^{m\times n}$ is given and $\langle\cdot,\cdot\rangle$ is the inner product considering X, Y as vectors with mn elements. We substitute X in the fidelity term with the auxiliary variable V to make the problem constrained form.

$$\arg\min_{X,V} \|X\|_{\mathbf{w},*} + \frac{\lambda}{2}\langle \log(\gamma^2 + (V-Y)^2), \mathbf{1}\rangle, \quad X = V \tag{4.21}$$

Now we introduce the corresponding augmented Lagrangian of (4.21) with Lagrangian multiplier $W \in \mathbb{R}^{m\times n}$ and penalty parameter $\beta > 0$.

$$\mathcal{L}_\beta(X,V,W) = \|X\|_{\mathbf{w},*} + \frac{\lambda}{2}\langle \log(\gamma^2 + (V-Y)^2), \mathbf{1}\rangle + \langle W, X-V\rangle + \frac{\beta}{2}\|X-V\|^2 \tag{4.22}$$

Then the nonconvex ADMM states that we can alternatively minimize the above Lagrangian $\mathcal{L}_\beta$ with respect to X and V to solve the given constrained problem (4.21). The corresponding algorithm is given in Algorithm 7.

Algorithm 7 Nonconvex-ADMM for (4.20)

Given a noisy Y.
Initialize $X^0,\ V^0,\ W^0$, set β.
for k=0,1,2,$\cdots$ **do**

$$X^{k+1} = \arg\min_X \|X\|_{\mathbf{w},*} + \frac{\beta}{2}\left\|X - V^k + \frac{W^k}{\beta}\right\|_2^2$$

$$V^{k+1} = \arg\min_V \frac{\lambda}{2}\left\langle \log\left(\gamma^2 + (V-Y)^2\right), \mathbf{1}\right\rangle + \frac{\beta}{2}\left\|X^{k+1} - V + \frac{W^k}{\beta}\right\|_2^2$$

$$W^{k+1} = W^k + \beta(X^{k+1} - V^{k+1})$$

if X^{k+1} satisfies the stopping criterion **then**
return X^{k+1} and stop
end if
end for

4.2.2 Convergence of algorithm 7

Following [52, 34], we can prove the convergence results of Algorithm 7 under suitable conditions.

Theorem 4.2.1. *Let $\{(X^k, V^k, W^k)\}_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 7. If $\beta > \dfrac{2\lambda}{\gamma^2}$, the sequence $\{(X^k, V^k, W^k)\}$ has at least one limit point and each limit point is a stationary point of $\mathcal{L}_\beta$.*

First, let us define the following:

$$\mathcal{F} : \mathbb{R}^{m\times n} \to \mathbb{R}, \quad \mathcal{F}(X) = \|X\|_{\mathbf{w},*}.$$
$$\mathcal{G} : \mathbb{R}^{m\times n} \to \mathbb{R}, \quad \mathcal{G}(X) = \frac{\lambda}{2}\langle \log(\gamma^2 + (X-Y)^2), \mathbf{1}\rangle.$$

To prove the theorem 4.2.1, we need the subgradient of weighted nuclear norm $\mathcal{F}$. A subgradient is defined on a convex function. But, in general, a weighted nuclear norm is nonconvex. Fortunately, according to [58, 30], we can get the subgradient of a weighted nuclear norm. For $X \in \mathbb{R}^{n\times m}$, let $U\Sigma V^T$ be the SVD of X and r be the rank of X. If we define the weight $\mathbf{w}$

as $[w_1, w_2, \ldots, w_l]$ where $l = \min(m, n)$, then the subgradient of $\mathcal{F}(X)$ is

$$\begin{aligned}\partial\mathcal{F}(X) =&\partial\|X\|_{\mathbf{w},*} = \{UW_rV^T + Z : Z \in \mathbb{R}^{m\times n},\\ &U^TZ = 0,\ ZV = 0,\ \sigma_i(Z) \le w_{r+i},\ i = 1, \ldots, l-r\}\end{aligned}$$

where W_r is diagonal matrix whose first r diagonal elements are $w_1, \ldots, w_r$ and all other elements are zero.

Lemma 4.2.2. *For all $k \ge 1$, there exists a constant $C_2 > 0$ and $p^{k+1} \in \partial\mathcal{L}_\beta(X^{k+1}, V^{k+1}, W^{k+1})$ such that $\|p^{k+1}\|_F^2 \le C_2\|V^k - V^{k+1}\|_F^2$.*

Proof. Here, we denote the partial derivatives of Lagrangian $\partial\mathcal{L}_\beta$ by

$$\partial\mathcal{L}_\beta(X^{k+1}, V^{k+1}, W^{k+1}) = (\partial_X\mathcal{L}_\beta, \nabla_V\mathcal{L}_\beta, \nabla_W\mathcal{L}_\beta)(X^{k+1}, V^{k+1}, W^{k+1}).$$

Note that $\partial_X\mathcal{L}_\beta$ requires the subgradient of $\mathcal{F}$.

By direct computation and (3.33),

$$\begin{aligned}\|\nabla_V\mathcal{L}_\beta(X^{k+1}, V^{k+1}, W^{k+1})\| &= \|W^k - W^{k+1}\| \le L_{\mathcal{G}}\|V^k - V^{k+1}\|.\\ \|\nabla_W\mathcal{L}_\beta(X^{k+1}, V^{k+1}, W^{k+1})\| &= \frac{1}{\beta}\|W^k - W^{k+1}\| \le \frac{L_{\mathcal{G}}}{\beta}\|V^k - V^{k+1}\|.\end{aligned}$$

Furthermore, observe that

$$\begin{aligned}\partial_X\mathcal{L}_\beta(X^{k+1}, V^{k+1}, W^{k+1}) &= \partial_X\mathcal{F}(X^{k+1}) + W^{k+1} + \beta(X^{k+1} - V^{k+1})\\ &= \partial_X\mathcal{F}(X^{k+1}) + W^k + \beta(X^{k+1} - V^k)\\ &\quad + (W^{k+1} - W^k) + \beta(V^k - V^{k+1}).\end{aligned}$$

From the optimality condition of line 4 in Algorithm 7, we have $0 \in \partial_X\mathcal{F}(X^{k+1}) + W^k + \beta(X^{k+1} - V^k)$. Therefore if we define p^{k+1} as follows:

$$p^{k+1} := \left((W^{k+1} - W^k) + \beta(V^k - V^{k+1}),\ W^k - W^{k+1},\ \frac{1}{\beta}(W^k - W^{k+1})\right),$$

then $p^{k+1} \in \partial\mathcal{L}_\beta(X^{k+1}, V^{k+1}, W^{k+1})$ and

$$\|p^{k+1}\| \le \left(L_{\mathcal{G}}(2 + \frac{1}{\beta}) + \beta\right)\|V^k - V^{k+1}\|,$$

where $C_2 = L_{\mathcal{G}}(2 + \frac{1}{\beta}) + \beta$. □

(Proof of Theorem 4.2.1). Like proof of the lemma 3.2.8, we can prove following inequality:

$$\mathcal{L}_\beta(X^{k+1}, V^k, W^k) - \mathcal{L}_\beta(X^k, V^k, W^k) \geq C_1 \|V^k - V^{k+1}\|_F^2 \tag{4.23}$$

and by the coercive of $\mathcal{F}$, we can prove the boundedness of $\{(X^k, V^k, W^k)\}$. Then we can take a subsequence $\{(X^{k_s}, V^{k_s}, W^{k_s})\}$ which converges to (X^*, V^*, W^*) as $s \to \infty$. And also, with the same logic as lemma 3.2.8, we can get that $\mathcal{L}_\beta(X^k, V^k, W^k)$ is non-increasing and lower-bounded. Therefore, $\mathcal{L}_\beta(X^k, V^k, W^k)$ converges. According to 4.23, $\|V^k - V^{k+1}\|_F^2$ converges to 0.

From lemma 4.2.2, there exists a sequence of subdifferentials $p^k \in \partial\mathcal{L}_\beta(X^k, V^k, W^k)$ that satisfies $\|p^k\| \to 0$ as $k \to \infty$. In particular, $\|p^{k_s}\| \to 0$ as $s \to \infty$. Because the roots of a polynomial depend continuously on its coefficients, $\mathcal{F}$ is continuous. This implies the continuity of $\mathcal{L}_\beta$ and $\lim_{s\to\infty} \mathcal{L}_\beta(X^{k_s}, V^{k_s}, W^{k_s}) = \mathcal{L}_\beta(X^*, V^*, W^*)$. Consequently, we have $0 \in \partial\mathcal{L}_\beta(X^*, V^*, W^*)$. □

4.2.3 Block matching method

Basically, we used the Frobenius norm to compute similarity of two patches. For noisy patches Y_i and Y_j, we use following metric:

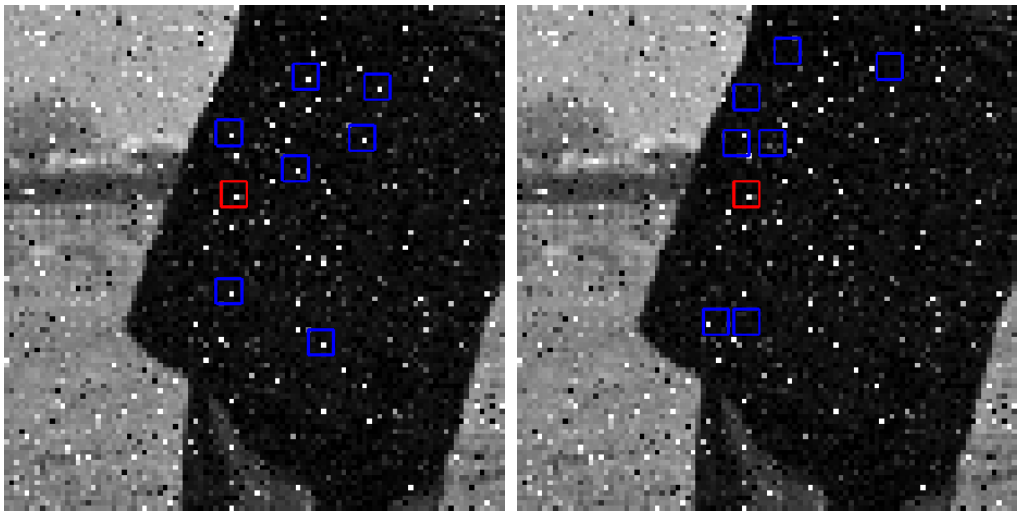
$$dist(Y_i, Y_j) = \|Y_i - Y_j\|_F^2$$

We set a reference patch by moving from the top left to the bottom right and find similar patches across the image (in practice, in a large local neighborhood of a reference patch). By sorting every distance between the reference patch and a neighbor patch, we stack the most similar k patches. But if we apply the Frobenius norm directly to the noisy image and find similar patches, it's hard to find similar patches with various noise. This phenomenon is due to the presence of very strong noise pixels that is a characteristic of Cauchy noise.

Figure 4.3 shows example of bad and good patch matching results. The red box represents a reference patch and the blue boxes represent matched

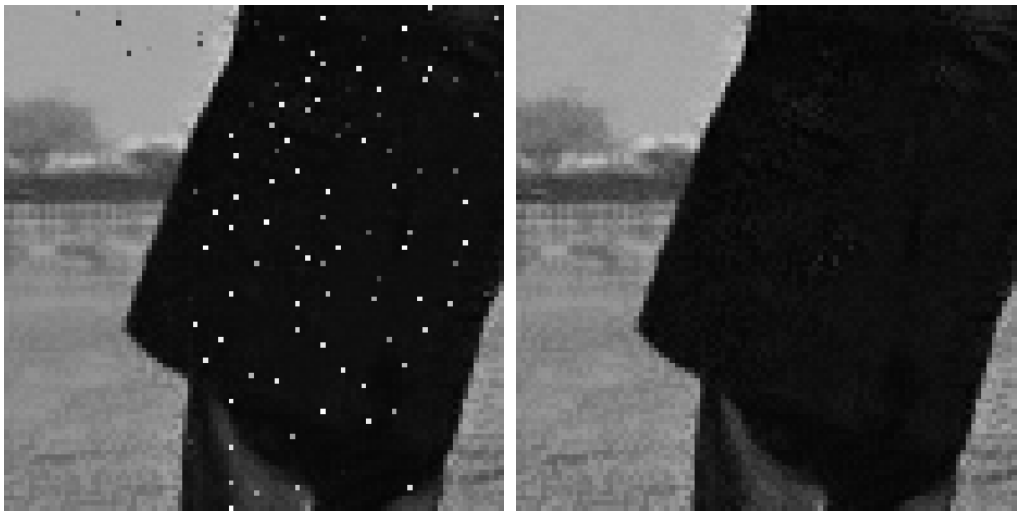
similar patches. Figure 4.3 (a) shows white dots in the blue boxes just like the white one in the red box. Here the white dots represent pixels contaminated with impulsive Cauchy noise. However, when white dots appear in the same location in the blue box, our algorithm do not recognize them as noise and do not erase them.

To fix this problem, we introduce a median filter which is particularly good for removing impulsive noise from a image. We applied 3 by 3 median filter to the noisy image then measured the similarity between the filtered reference patch and a filtered neighbor patch by using the Frobenius norm. Figure 4.3 (b) shows the result of similar patches found with the Frobenius norm after applying a median filter to the noisy image. The blue boxes in figure 4.3 (b) have various noisy patterns that give good noise removal results. Figure 4.4 (a) and (b) show the denoising results of the bad and good patch matching in figure 4.3 (a) and (b), respectively. We can see that noise remains in figure 4.4 (a), and that noise has been clearly removed in figure 4.4 (b).



(a) Bad patch matching (b) Good patch matching

Figure 4.3: Example of patch matching results



(a) Bad patch matching result (b) Good patch matching result

Figure 4.4: Example of denoising result of bad and good patch matching

4.3 Numerical Results Of Weighted Nuclear Norm Denoising Model For Cauchy Noise

We present the results of several experiments to demonstrate the performance of the proposed algorithm for Cauchy noise removal. We uploaded the code for our algorithm on `https://github.com/sbob25/CauchynoiseWNNM`. We use twelve 256×256 gray-scale test images for our experiments, as shown in Figure 3.1 in chapter 3. As in chapter 3, for the original image u_0, we generate the noisy image y corrupted by Cauchy noise n which follows the Cauchy distribution $\mathcal{C}(0, \gamma)$:

$$y = u_0 + n = u_0 + \gamma \frac{\eta_1}{\eta_2},$$

by using proposition 2.1.9. We test the noise intensity $\gamma = 5$ and 10. For qualitative measures of the restored image, we adopt the PSNR and the SSIM which are defined in chapter 3.

4.3.1 Parameter setting and truncated weighted nuclear norm

In the noisy image y, we set the size of a noisy reference patch y_j as 6×6. Then, for a noisy reference patch y_j, we make a matrix Y_j which is consist of 70 similar patches of y_j by searching within the range of 30×30 from the reference patch. Also, we change a reference patch from top left to bottom right with a stride of 4. And for each Y_j, we minimize the model:

$$\arg\min_{X_j} \|X_j\|_{\mathbf{w},*} + \frac{\lambda}{2}\langle \log(\gamma^2 + (X_j - Y_j)^2), \mathbf{1}\rangle. \tag{4.24}$$

To solve the above model, we have to set the weight $\mathbf{w}$. In [22], Gu et al. used the prior knowledge that the larger singular values of X_j have more important data than smaller ones and so defined weight as follows:

$$w_i = \frac{c\sqrt{n}}{\sigma_i(X_j) + \epsilon}$$

where $c > 0$ is a constant, n is a patch size y_j, and $\epsilon = 10^{-15}$ is a small value to prevent dividing by zero.

In Cauchy noise removal, we found that even small singular values have some important information. Therefore we define the weight as follows by introducing an maximum value Θ:

$$w_i = \min\left(\frac{c\Theta^2\sqrt{n}}{\sigma_i(\mathbf{Y}) + \epsilon},\ \Theta^2\right),$$

i.e. a weight w_i cannot be bigger than the Θ^2.

To make the sequence $\{(X^k, V^k, W^k)\}_{k\in\mathbb{N}}$ converge in the algorithm 7, we set β to be slightly larger than $\dfrac{2\lambda}{\gamma^2}$. And we set $\lambda = 1$, $c = 23$, $\Theta = 2.7$ for $\gamma = 5$ and $\Theta = 4$ for $\gamma = 10$ in our experiments.

Initialization is another issue because our model is non-convex. We obtained a starting image $x^{(0)}$ by eliminating noisy pixels that have extreme values. We applied a 3×3 median filter to a noisy image y to obtain a filtered image $\tilde{y}$ and truncated y by the minimum and maximum of all pixel values

of $\tilde{y}$. In other words,

$$x^{(0)} = \max(m, \min(M, y)),$$

where m, M are minimum and maximum among all pixel values of $\tilde{y}$ respectively.

As we did not conduct an exhaustive search for other possible choices of parameters such as patch size and number of stacked patches, the performance of our algorithm can be further improved.

4.3.2 Termination condition

We define an energy function to be utilized for setting a stopping criterion of our algorithm. Because the weight of the weighted nuclear norm varies with respect to iteration, it is inappropriate to set the termination condition using the objective function (4.19). Instead, we use the nuclear norm for the energy function E:

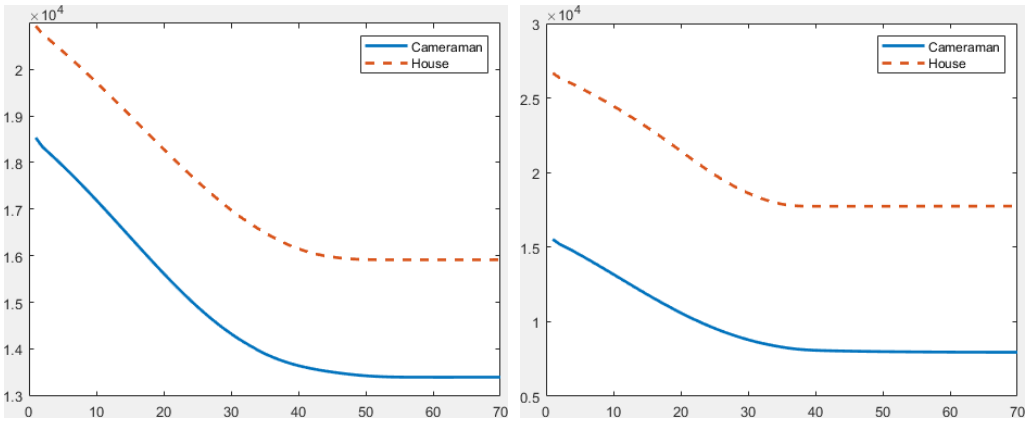
$$E(X) = \|X\|_* + \frac{\lambda}{2} \sum_{x \in X, y \in Y} \log\left(\gamma^2 + (x - y)^2\right). \tag{4.25}$$

where X is a stacked matrix of flattened similar patches of the reference patch in the restored image and Y is the corresponding matrix whose elements are selected from the noisy image y. Theoretically, we have to compute $E(X)$ for every reference patch in the image. However, this requires high computational cost; therefore, we compute E only for the patch $\hat{p}$, which has the smallest standard deviation among the patches of filtered image `medfilt2`(f).

Subsequently, we terminate the algorithm at iteration k when the criterion

$$\frac{|E(X^{(k-1)}) - E(X^{(k)})|}{|E(X^{(k)})|} < \epsilon$$

is satisfied for a threshold ϵ. In the experiments, we set $\epsilon = 10^{-3}$. Figure 4.5 shows progress of the energy (4.25) of the cameraman and the house image. As you can see, the energies decrease well according to the iteration, so these are suitable for use as termination condition.



(a) (b)

Figure 4.5: Plots of the energy values versus iterations for the noisy images "Cameraman" and "House" with (a) $\gamma = 5$ and (b) 10.

4.3.3 Experimental results

We compare our nonconvex WNNM model with CTV [45], NCTV [34], and Myriad [28]. Table 4.1 shows the results of CTV, NCTV, Myriad and ours with noise intensity $\gamma = 5$. In table 4.1, we can see that the results of our model are better than the others in both PSNR and SSIM for all 12 images. The PSNR values of our WNNM model are about 1.9 higher than that of the others on average. And the average SSIM value difference between our WNNM and the others is about 0.04. Figure 4.6 shows that our WNNM model has better performance than the others without PSNR or SSIM comparison. If you look at the back and arms of the person behind of the man in figure 4.6, you can see that our WNNM model are good at removing noise, but CTV, NCTV and Myriad are making blur or remaining noise. Also the men's sleeve are recovered very clearly in our WNNM model result, unlike the others. See the figure 4.7. Especially, our WNNM model is effective to remove noise of texture. Figure 4.8 and figure 4.9 show the textures of the tablecloth and the woman's cloths have been improved most sharply in our WNNM result. Looking at the peppers in figure 4.10, you see that our WNNM model removes noise clearly, while the others leave noise. Figure 4.11 shows an zoomed view of the top of the bell pepper in figure 4.10. Our

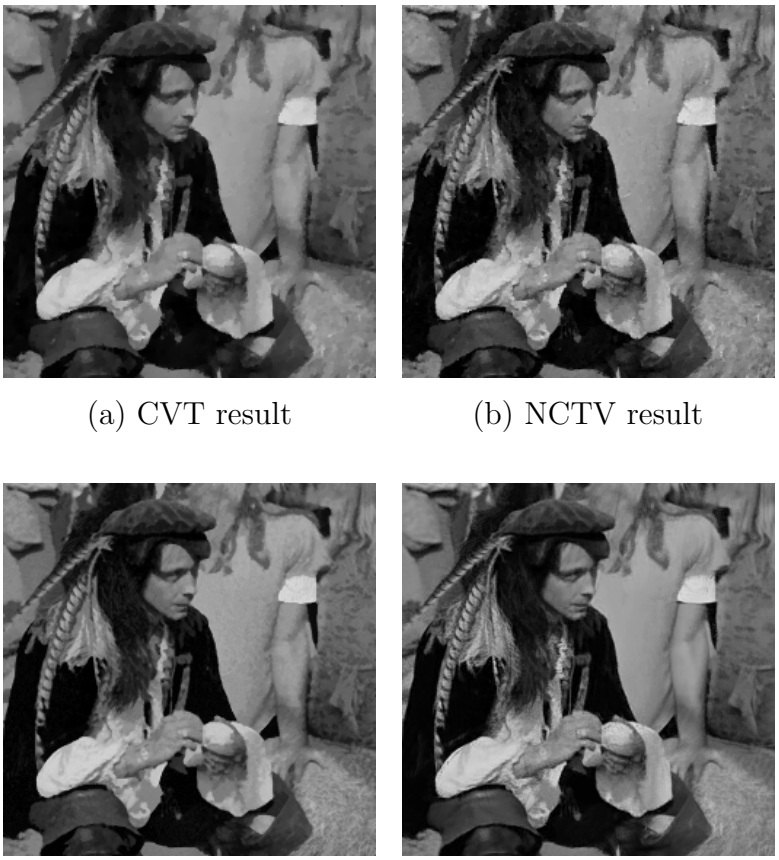
WNNM model, unlike the other models, removes well the noise at the top of the bell pepper and does not lose clearity.

	PSNR					SSIM				
	Noisy	CTV	NCTV	Myriad	Ours	Noisy	CTV	NCTV	Myriad	Ours
Barbara	19.21	29.41	29.49	30.98	**33.17**	0.4265	0.8500	0.8533	0.8871	**0.9306**
Boat	19.19	29.45	29.92	29.53	**30.91**	0.4058	0.8519	0.8693	0.8523	**0.8905**
Cameraman	19.23	27.97	28.36	28.96	**31.00**	0.3563	0.8571	0.8482	0.8376	**0.9001**
Couple	19.19	28.54	29.55	29.34	**30.92**	0.4243	0.8190	0.8650	0.8557	**0.9004**
House	19.13	29.04	29.33	29.55	**30.49**	0.4333	0.8705	0.8775	0.8699	**0.9092**
Lena	19.19	30.32	31.24	31.68	**33.44**	0.3572	0.8721	0.8851	0.8799	**0.9330**
Man	19.20	28.67	28.84	28.83	**29.85**	0.4744	0.8422	0.8604	0.8582	**0.8781**
Mandrill	19.21	26.46	27.00	27.92	**28.40**	0.5000	0.7499	0.8117	0.8332	**0.8525**
Peppers	19.24	30.92	31.00	30.67	**32.50**	0.3859	0.8929	0.8854	0.8712	**0.9166**
Plane	19.20	30.23	30.47	30.05	**31.37**	0.3727	0.9021	0.8896	0.8654	**0.9228**
Synthetic1	19.21	39.36	41.51	35.68	**43.81**	0.1498	0.9609	0.9728	0.8394	**0.9833**
Synthetic2	19.25	36.26	36.32	35.47	**43.89**	0.1896	0.9274	0.8995	0.8278	**0.9852**

Table 4.1: PSNR and SSIM for the noisy and restored images by different methods ($\gamma = 5$)

	PSNR					SSIM				
	Noisy	CTV	NCTV	Myriad	Ours	Noisy	CTV	NCTV	Myriad	Ours
Barbara	16.31	27.57	27.81	27.70	**30.48**	0.2871	0.7963	0.8027	0.7665	**0.8758**
Boat	16.30	27.25	27.75	27.11	**28.86**	0.2738	0.7789	0.7944	0.7245	**0.8399**
Cameraman	16.33	26.40	26.86	26.86	**29.04**	0.2427	0.8039	0.7858	0.6910	**0.8627**
Couple	16.32	26.88	27.27	26.73	**28.64**	0.2834	0.7567	0.7814	0.7270	**0.8378**
House	16.26	26.78	27.19	26.83	**28.40**	0.3026	0.8022	0.8072	0.7424	**0.8631**
Lena	16.33	28.84	28.53	28.38	**30.54**	0.2361	0.8446	0.8385	0.7408	**0.8879**
Man	16.31	26.52	26.96	26.50	**27.73**	0.3227	0.7627	0.7899	0.7515	**0.8049**
Mandrill	16.33	24.67	25.23	25.58	**26.27**	0.3416	0.6279	0.7164	0.7175	**0.7513**
Peppers	16.36	28.69	29.00	27.79	**30.49**	0.2629	0.8504	0.8511	0.7419	**0.8866**
Plane	16.31	27.86	28.21	27.67	**29.36**	0.2564	0.8528	0.8250	0.7235	**0.8924**
Synthetic1	16.33	36.48	38.15	30.85	**41.39**	0.0765	0.9420	0.9523	0.6187	**0.9836**
Synthetic2	16.34	35.56	36.29	30.24	**40.49**	0.1016	0.9361	0.9400	0.5998	**0.9811**

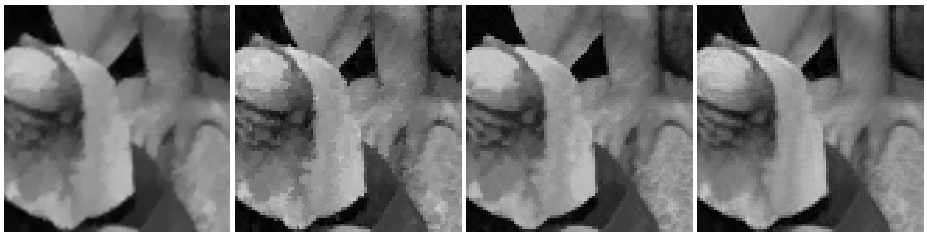
Table 4.2: PSNR and SSIM for the noisy and restored images by different methods ($\gamma = 10$)



(a) CVT result (b) NCTV result

(c) Myriad result (d) Our WNNM result

Figure 4.6: Man image results of several Cauchy denoising models ($\gamma = 5$)



(a) CTV (b) NCTV (c) Myriad (d) Ours

Figure 4.7: Zoomed of 4.6

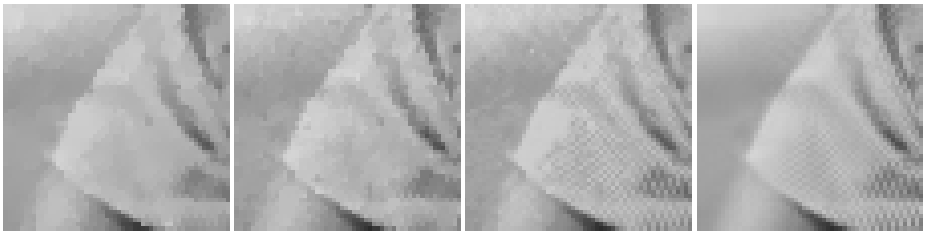
(a) CVT result

(b) NCTV result

(c) Myriad result

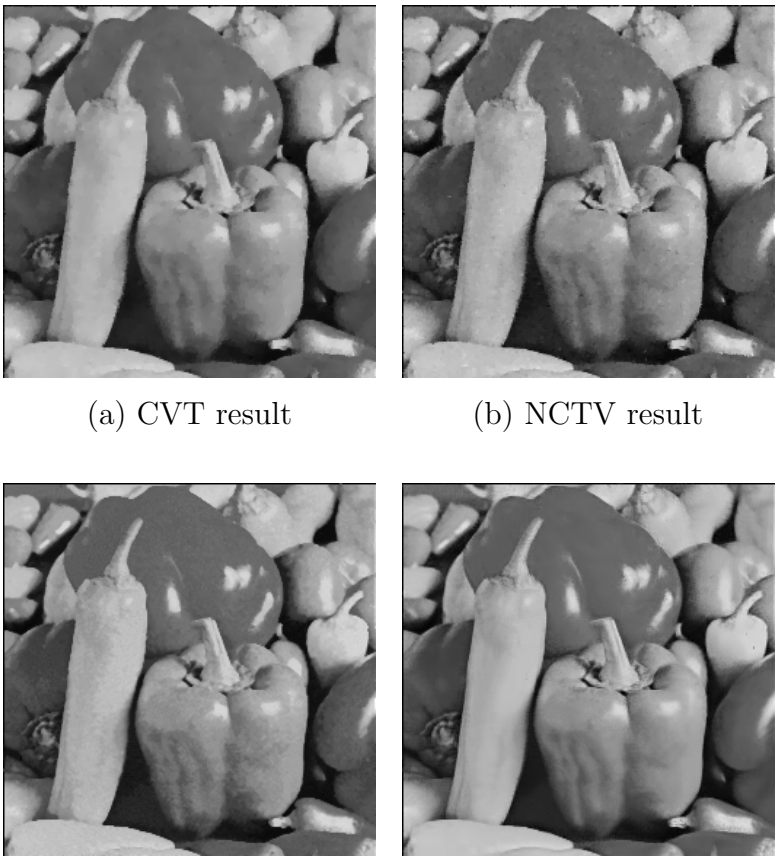
(d) Our WNNM result

Figure 4.8: Barbara image results of several Cauchy denoising models ($\gamma = 5$)



(a) CTV (b) NCTV (c) Myriad (d) Ours

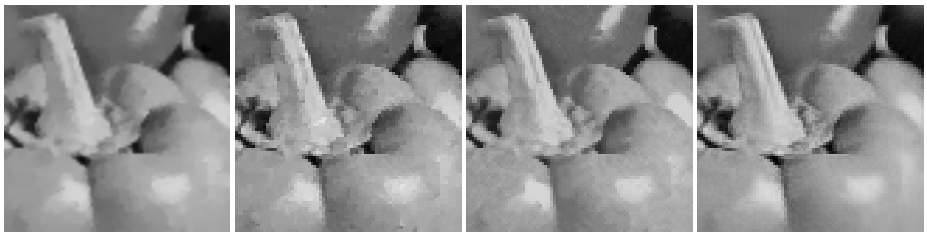
Figure 4.9: Zoomed of 4.6



(a) CVT result (b) NCTV result

(c) Myriad result (d) Our WNNM result

Figure 4.10: Pepper image results of several Cauchy denoising models ($\gamma = 5$)



(a) CTV (b) NCTV (c) Myriad (d) Ours

Figure 4.11: Zoomed of 4.10

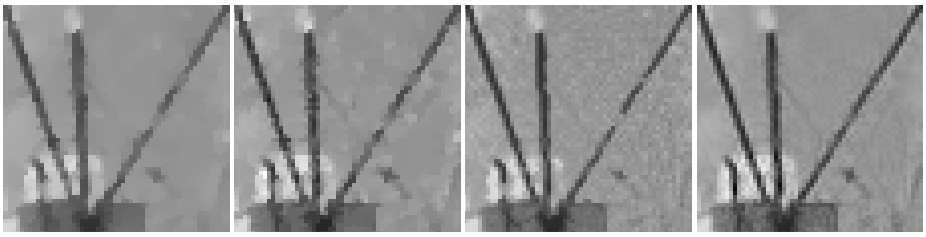
(a) CVT result

(b) NCTV result

(c) Myriad result

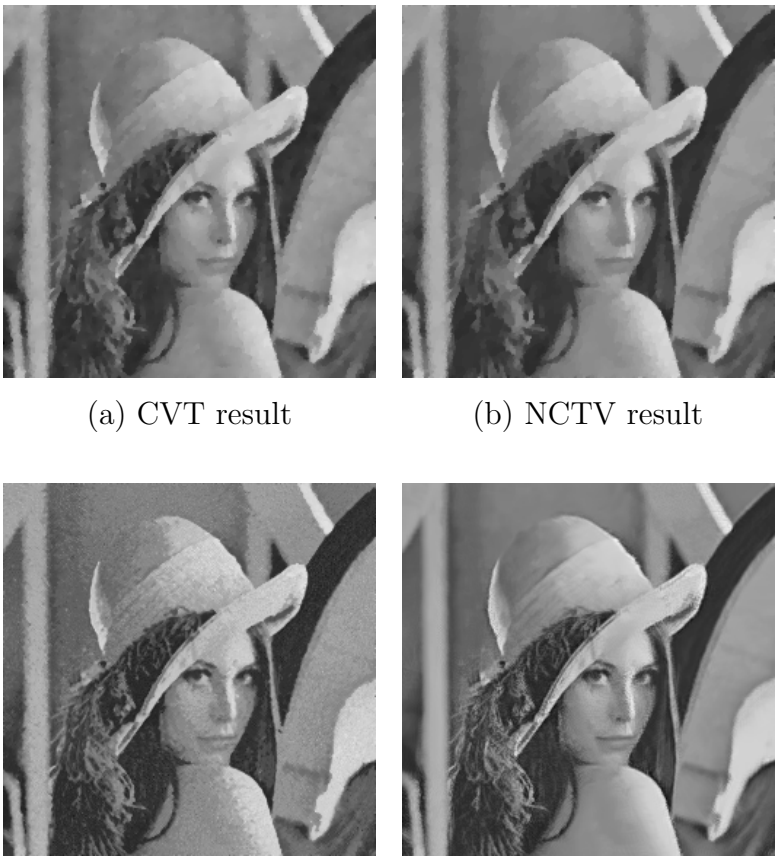
(d) Our WNNM result

Figure 4.12: Boat image results of several Cauchy denoising models ($\gamma = 10$)



(a) CTV (b) NCTV (c) Myriad (d) Ours

Figure 4.13: Zoomed of 4.12



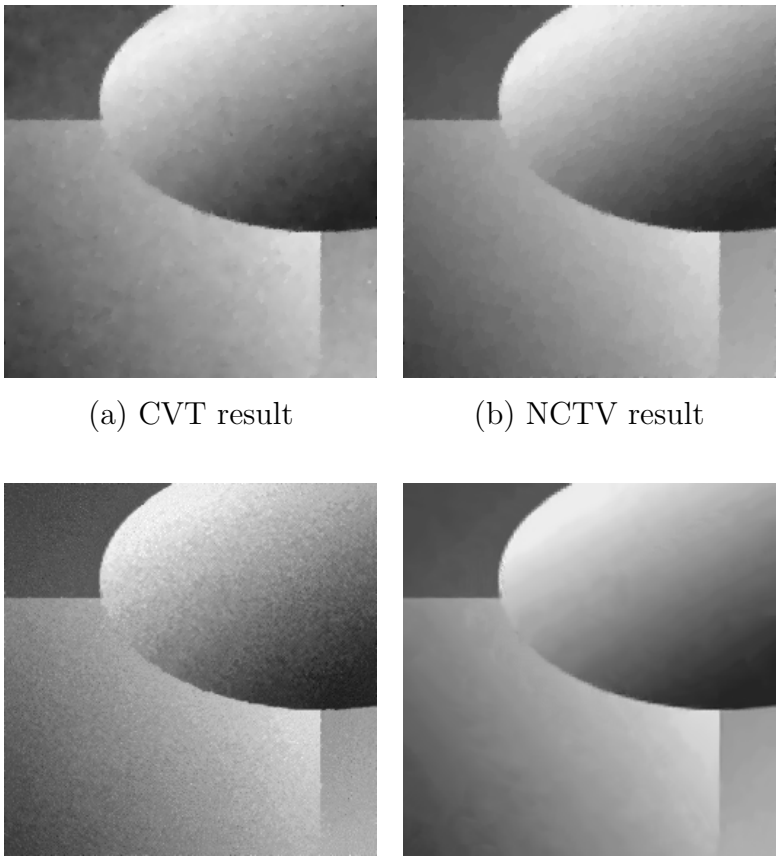
(a) CVT result (b) NCTV result

(c) Myriad result (d) Our WNNM result

Figure 4.14: Lena image results of several Cauchy denoising models ($\gamma = 10$)

(a) CTV (b) NCTV (c) Myriad (d) Ours

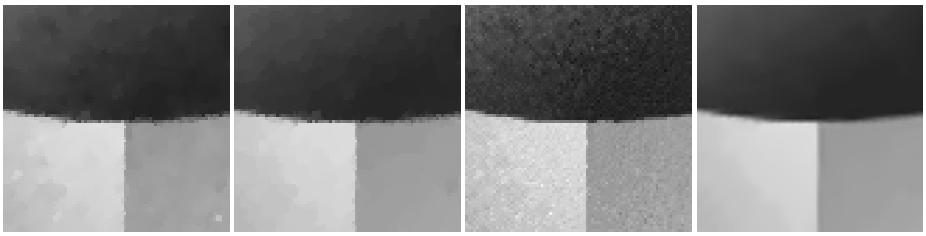
Figure 4.15: Zoomed of 4.14



(a) CVT result (b) NCTV result

(c) Myriad result (d) Our WNNM result

Figure 4.16: Synthetic1 image results of several Cauchy denoising models ($\gamma = 10$)



(a) CTV (b) NCTV (c) Myriad (d) Ours

Figure 4.17: Zoomed of 4.16

For the case of heavy noise $\gamma = 10$, table 4.2 shows that the results of our WNNM model have the highest PSNR and SSIM values in all images. On average, the PSNR values of our WNNM model and the others differ by about 1.8 and the differences of SSIM values of our WNNM and the others are about 0.05. For the synthetic2 image, PSNR values of our WNNM model are 4 or more higher than that of the others. Also the Mandrill image has the least PSNR difference between our model and the others. Our model has about 0.5 higher value than the others for the Mandrill image results. The boat image denoising results for $\gamma = 10$ shown in figure 4.12. The (a) CVT result were over-smoothed, and (b) NCTV and (c) Myriad results show remaining noise. But (d) our WNNM model result preserves edges well while at the same time removing noise as well. Figure 4.14 shows denoising results of the Lena image of several models for noise intensity $\gamma = 10$. Lena's face recovered by our model is more clear than that of the others. And our model recovered well the texture of fur on the hat without blurring. Finally, we compare the result of the Synthetic1 image (figure 4.16). As you see, our model removed noise clearly and recovered the surface and the edges well unlike the other models.

Chapter 5

Conclusion

In this thesis, we studied about Cauchy noise and Cauchy noise removal by variational models. We presented two models. First model was an local approach based on fractional TV and second thing used an non-local method which is based on weighted nuclear norm. For both models, we applied nonconvex ADMM and proved the convergence of a nonconvex ADMM sequence to an stationary point of the augmented Lagrangian.

Fractional TV model is almost similar with the conventional TV in terms of computational speed. And the fractional TV model had better denoising performance than the conventional TV model. In particular, increasing the volume of the recovered images as the alpha increases shows that it is worth applying fractional TV. But, because of the similarity of models, the difference of performance was not much significant. And for heavy noise $\gamma = 10$, as α increased, round artifacts appeared in recovered images, which were unsatisfactory results. To improve these disadvantages, future work might lead to the study of various generalization of total variation model (Total generalized variation [4], Structure tensor total variation [29], etc.).

The second our model, the WNNM denoising model for Cauchy noise, was excellent enough to yield much better results than state-of-the-art Cauchy noise removal models. We used median filter and ratio of total variation to find good similar patches which are showed good results, but the results was still worse than using the clean (original) image for patch matching. Further research is expected to find better patch matching methods. Another

important issue how to define weights of the weighted nuclear norm. In the original WNNM model, Gu et al. define weights as the reciprocal of singular values. But in Cauchy noise removal, we have found that truncated weights are much better performance than Gu's suggested weights. Thus, if we study more about weights, we may find weights that are more optimal for Cauchy noise. Finally, the computation speed is rather slow because our model is non-local. To improve the speed, we tested several parameter values and found the values that speed up without loss of recover quality and used parallel computing. But it is still not fast. The computation using GPU is expected to speed up and our model will be able to be used effectively even in the removal of Cauchy noise in the industrial site.

Bibliography

[1] Alin Achim and Ercan E Kuruoglu. Image denoising using bivariate α-stable distributions in the complex wavelet domain. *IEEE Signal Processing Letters*, 12(1):17–20, 2004.

[2] Om P Agrawal. Fractional variational calculus in terms of riesz fractional derivatives. *Journal of Physics A: Mathematical and Theoretical*, 40(24):6287, 2007.

[3] E. Artin. *The gamma function*. Athena Series : Selected Topics in Mathematics. Holt, Rinehart and Winston, 1964.

[4] Kristian Bredies, Karl Kunisch, and Thomas Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[5] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[6] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

[7] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted ℓ 1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

[8] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[9] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[10] Dali Chen, YangQuan Chen, and Dingyu Xue. Fractional-order total variation image denoising based on proximity algorithm. *Applied Mathematics and Computation*, 257:537–545, 2015.

[11] Mandar A Chitre, John R Potter, and S-H Ong. Optimal and near-optimal signal detection in snapping shrimp dominated ambient noise. *IEEE Journal of oceanic engineering*, 31(2):497–503, 2006.

[12] J. H. Curtiss. On the distribution of the quotient of two chance variables. *Ann. Math. Statist.*, 12(4):409–421, 12 1941.

[13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[14] Geoff Davis, Stephane Mallat, and Marco Avellaneda. Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98, 1997.

[15] Hamza El Ghannudi, Laurent Clavier, Nourddine Azzaoui, François Septier, and Paul-Alain Rolland. α-stable interference modeling and cauchy receiver for an ir-uwb ad hoc network. *IEEE Transactions on Communications*, 58(6):1748–1757, 2010.

[16] G. B Folland. *Advanced calculus*. Upper Saddle River, NJ : Prentice Hall, 2002. Includes bibliographical references (p. 453-454) and index.

[17] Catherine Forbes, Merran Evans, Nicholas Hastings, and Brian Peacock. *Statistical distributions*. John Wiley & Sons, 2011.

[18] Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.

[19] Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343, 2009.

[20] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[21] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International journal of computer vision*, 121(2):183–208, 2017.

[22] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.

[23] Abdalla Mohamed Hambal, Dr. Zhijun Pei, and Faustini Libent Ishabailu. Image noise reduction and filtering techniques. 2017.

[24] J. Hou, L. Chau, N. Magnenat-Thalmann, and Y. He. Sparse low-rank matrix approximation for data compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(5):1043–1054, May 2017.

[25] Moshe Idan and Jason L Speyer. Cauchy estimation for linear scalar systems. *IEEE transactions on automatic control*, 55(6):1329–1342, 2010.

[26] Saleem A Kassam. *Signal detection in non-Gaussian noise*. Springer Science & Business Media, 2012.

[27] Ercan E Kuruoglu, William J Fitzgerald, and Peter JW Rayner. Near optimal detection of signals in impulsive noise modeled with a symmetric/spl alpha/-stable distribution. *IEEE Communications Letters*, 2(10):282–284, 1998.

[28] Friederike Laus, Fabien Pierre, and Gabriele Steidl. Nonlocal myriad filters for cauchy noise removal. *Journal of Mathematical Imaging and Vision*, 60(8):1324–1354, 2018.

[29] Stamatios Lefkimmiatis, Anastasios Roussos, Petros Maragos, and Michael Unser. Structure tensor total variation. *SIAM Journal on Imaging Sciences*, 8(2):1090–1122, 2015.

[30] Yu-Fan Li, Yan-Jiao Zhang, and Zheng-Hai Huang. A reweighted nuclear norm minimization algorithm for low rank matrix recovery. *Journal of Computational and Applied Mathematics*, 263:338–350, 2014.

[31] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *ICML*, volume 1, page 8, 2010.

[32] Artur Loza, David Bull, Nishan Canagarajah, and Alin Achim. Non-gaussian model-based fusion of noisy images in the wavelet domain. *Computer Vision and Image Understanding*, 114(1):54–65, 2010.

[33] Ivan Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008.

[34] Jin-Jin Mei, Yiqiu Dong, Ting-Zhu Huang, and Wotao Yin. Cauchy noise removal by nonconvex admm with convergence guarantees. *Journal of Scientific Computing*, 74(2):743–766, 2018.

[35] Christopher A Metzler, Arian Maleki, and Richard G Baraniuk. Bm3d-amp: A new image recovery algorithm based on bm3d denoising. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3116–3120. IEEE, 2015.

[36] Charles A Micchelli, Lixin Shen, and Yuesheng Xu. Proximity algorithms for image models: denoising. *Inverse Problems*, 27(4):045009, 2011.

[37] Leon Mirsky. A trace inequality of john von neumann. *Monatshefte für mathematik*, 79(4):303–306, 1975.

[38] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[39] John Nolan. Stable distribution: Models for heavy-tailed data. 01 2014.

[40] Yijin Peng, Jiayu Chen, Xin Xu, and Fangling Pu. Sar images statistical modeling and classification based on the mixture of alpha-stable distributions. *Remote Sensing*, 5(5):2145–2163, 2013.

[41] Igor Podlubny. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, volume 198. Elsevier, 1998.

[42] Igor Podlubny, Aleksei Chechkin, Tomas Skovranek, YangQuan Chen, and Blas M. Vinagre Jara. Matrix approach to discrete fractional calculus ii: Partial fractional differential equations. *Journal of Computational Physics*, 228(8):3137 – 3153, 2009.

[43] Paul M Reeves. A non-gaussian turbulence simulation. Technical report, WASHINGTON UNIV SEATTLE DEPT OF AERONAUTICS AND ASTRONAUTICS, 1969.

[44] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[45] Federica Sciacchitano, Yiqiu Dong, and Tieyong Zeng. Variational approach for restoring blurred images with cauchy noise. *SIAM Journal on Imaging Sciences*, 8(3):1894–1922, 2015.

[46] Guifang Shao, Yubo Gao, Jiayu Zuo, Yuqing Yue, and Jianbo Yang. An improved bm3d method for edna mieroarray image denoising. In *2018 13th International Conference on Computer Science & Education (ICCSE)*, pages 1–6. IEEE, 2018.

[47] Nathan Srebro and Ruslan R Salakhutdinov. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*, pages 2056–2064, 2010.

[48] GA Tsihrintzis, P Tsakalides, and CL Nikias. Signal detection in severely heavy-tailed radar clutter. In *Conference Record of The Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 865–869. IEEE.

[49] Siva Ram Krishna Vadali, Priyadip Ray, Subrahmanyam Mula, and Pramod K Varshney. Linear detection of a weak signal in additive cauchy noise. *IEEE Transactions on Communications*, 65(3):1061–1076, 2017.

[50] Fenghui Wang, Wenfei Cao, and Zongben Xu. Convergence of multi-block bregman admm for nonconvex composite problems. *Science China Information Sciences*, 61(12):122101, 2018.

[51] Yilun Wang, Wotao Yin, and Yin Zhang. A fast algorithm for image deblurring with total variation regularization. 01 2007.

[52] Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

[53] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[54] Pierre Weiss, Laure Blanc-Féraud, and Gilles Aubert. Efficient schemes for total variation minimization under constraints in image processing. *SIAM journal on Scientific Computing*, 31(3):2047–2080, 2009.

[55] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–36, 2018.

[56] Lei Yang, Ting Kei Pong, and Xiaojun Chen. Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction. *SIAM Journal on Imaging Sciences*, 10(1):74–110, 2017.

[57] Yuh-Chin Chang, S. R. Kadaba, P. C. Doerschuk, and S. B. Gelfand. Image restoration using recursive markov random field models driven by cauchy distributed noise. *IEEE Signal Processing Letters*, 8(3):65–66, March 2001.

[58] Zhiyuan Zha, Xin Yuan, Bei Li, Xinggan Zhang, Xin Liu, Lan Tang, and Ying-Chang Liang. Analyzing the weighted nuclear norm minimization and nuclear norm minimization based on group sparse representation. *arXiv preprint arXiv:1702.04463*, 2017.

[59] Jianping Zhang and Ke Chen. A total fractional-order variation model for image restoration with nonhomogeneous boundary conditions and its numerical solution. *SIAM Journal on Imaging Sciences*, 8(4):2487–2518, 2015.

[60] Xiaoqun Zhang, Martin Burger, Xavier Bresson, and Stanley Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3):253–276, 2010.

[61] Manfred Zimmermann and Klaus Dostert. Analysis and modeling of impulsive noise in broad-band powerline communications. *IEEE transactions on Electromagnetic compatibility*, 44(1):249–258, 2002.

BIBLIOGRAPHY

국문초록

이미지 처리에서 이미지 잡음 제거는 가장 중요한 문제 중 하나다. 이 논문에서 우리는 다양한 접근 방식에 의한 코시 잡음 제거를 연구한다. 코시 잡음은 엔지니어링 애플리케이션에서 자주 발생하나 코시 잡음의 변분법적 모델의 비 볼록성으로 인해 해결하기가 어렵고 많이 연구되지 않았다. 코시 노이즈를 제거하기 위해 우리는 곱셈기의 볼록하지 않은 교류 방향 방법(nonconvex ADMM)을 사용하였으며 두 가지 변분법적 모델을 제시한다. 첫 번째는 분수 총 변이(FTV)를 이용한 모델이다. 분수 총 변이는 일반적인 정수 도함수를 실수 도함수로 확장 한 분수 도함수에 의해 정의된다. 두 번째는 가중 핵 노름을 이용한 모델이다. 가중 핵 노름은 저수준 영상처리에서 탁월한 성능을 발휘한다. 우리는 가중 핵 노름이 코시 잡음 제거에서도 뛰어난 성능을 발휘할 것으로 예상하였고, 우리의 새로운 아이디어를 가중 핵 노름 최소화와 결합하여 현존하는 코시 잡음 제거 최신 모델들보다 더 나은 결과를 얻을 수 있었다. 마지막 장에서 실제 코시 잡음 제거 테스트를 통해 우리 모델이 얼마나 뛰어난지 확인하며 논문을 마친다.

주요어휘**:** 코시 잡음, 영상 노이즈제거, 비볼록 승수의 교대방향 방법, 분수적 총변이, 가중 핵 노름, 변분법적 모델
학번**:** 2012-23017